SEW EURODRIVE

Handbuch

MultiMotion / MultiMotion Light

Technologie-Modul

Kinematics

Ausgabe 07/2014

20211732 / DE

Inhaltsverzeichnis

1 Allgemeine Hinweise 7
1.1 Gebrauch der Dokumentation 7
1.2 Aufbau der Sicherheitshinweise 7
1.2.1 Bedeutung der Signalworte 7
1.2.2 Aufbau der abschnittsbezogenen Warnhinweise 7
1.2.3 Aufbau der eingebetteten Warnhinweise 8
1.3 Mängelhaftungsansprüche 8
1.4 Haftungsausschluss 9
1.5 Mitgeltende Unterlagen 9
1.6 Produktnamen und Marken 9
1.7 Urheberrechtsvermerk 9

2 Sicherheitshinweise 10
2.1 Allgemein 10
2.2 Zielgruppe 10
2.3 Bestimmungsgemäße Verwendung 11
2.4 Bussysteme 11

3 Systembeschreibung 12
3.1 Inhalt dieses Handbuchs 12
3.1.1 Schreibkonventionen 13
3.2 Voraussetzungen 13
3.2.1 Hardware 13
3.2.2 Software 13
3.2.3 Erforderliche Technologiestufe 13
3.3 Funktionen 15
3.4 Anwendungsbereiche 17

4 Grundlagen der Robotik 18
4.1 Bahnsteuerung 18
4.2 Arten der Kinematiktransformation 18
4.3 Beispiele für weitere Transformationen 18
4.4 Kartesische Position 19

5 Auswahl von Kinematics oder HandlingKinematics 21
5.1 Ebene 1: Applikationsmodul HandlingKinematics für CCU 21
5.1.1 Eigenschaften 21
5.1.2 Vorteile 21
5.2 Ebene 2: Technologiemodul HandlingKinematics für MOVI-PLC® 22
5.2.1 Eigenschaften 22
5.2.2 Zusätzliche Vorteile 22
5.3 Ebene 3: Technologiemodul Kinematics für MOVI-PLC® 23
5.3.1 Eigenschaften 23
5.3.2 Zusätzliche Vorteile 23

6 Inbetriebnahme und Konfiguration 25
6.1 Ablauf 25
6.2 Voraussetzungen 26

20211732 / DE – 07/2014

6.3 Schritt 1: Erstellung eines MotionStudio-Projekts ... 26
6.4 Schritt 2: Erstellung eines MOVI-PLC®-Projekts ... 27
6.5 Schritt 3: Integration des Technologiemoduls ... 28
6.6 Schritt 4: Einzelachsen in Betrieb nehmen ... 30
6.7 Schritt 5: Konfiguration der Kinematik ... 30
6.7.1 Neues Technologiemodul einfügen ... 30
6.7.2 Auswahl von "Kinematics" oder "HandlingsKinematics" ... 32
6.7.3 Auswahl Gerätetyp und Kinematikmodell ... 33
6.7.4 Konfiguration der Achsen ... 34
6.7.5 Anpassung der Modellachsen an reale Achsen ... 35
6.7.6 Einstellung der Kinematikparameter ... 36
6.7.7 Einstellung der Kinematischen Software-Endschalter ... 36
6.7.8 Kartesische Einstellungen ... 37
6.7.9 ContinuousPath-Einstellungen ... 38
6.7.10 Weitere Einstellungen ... 39
6.7.11 Standardbelegung von Eingangsvariablen ... 39
6.7.12 Download ... 40

7 Diagnose ... 41
7.1 Kinematikmonitor ... 41
7.1.1 Bedienoberfläche ... 41
7.1.2 Monitor- und Steuerbetrieb ... 43
7.1.3 Einstellung für kinematische Betriebsart ... 45
7.1.4 Referenzieren der Achsen ... 46
7.1.5 Achsweiser Tippbetrieb ... 46
7.1.6 Kartesischer Tippbetrieb ... 47
7.1.7 Achsweise TARGET-Betriebsart ... 48
7.1.8 Kartesische TARGET-Betriebsart ... 49
7.1.9 ContinuousPath-Betriebsart ... 49
7.1.10 Wahl der Transformation ... 50
7.1.11 Verschiedene Einstellungen ... 52
7.2 3D-Simulation ... 53
7.2.1 Zweck ... 53
7.2.2 Voraussetzungen ... 53
7.2.3 Starten der 3D-Simulation ... 54
7.2.4 Anpassen der 3D-Simulation ... 54
7.3 MessageHandler ... 55

8 Kinematikmodelle ... 57
8.1 CARTESIAN GANTRY ... 57
8.2 ROLLER GANTRY ... 60
8.3 SCARA ... 62
8.4 DELTA ... 64
8.5 TRIPOD ... 65
8.6 MIXED ... 66
8.7 USER-Kinematik ... 66

9 Interpolierende Betriebsarten ... 67

20211732 / DE – 07/2014

9.1 Überblick ... 67
9.2 Tippbetrieb ... 68
9.2.1 Vorgabe der Richtung und kinematischer Bewegungsgrößen ... 68
9.3 TARGET-Betriebsarten ... 68
9.3.1 Prinzip ... 68
9.3.2 Beispiel: Pick-and-Place ... 69
9.3.3 Vorteile ... 70
9.3.4 KIN_TARGET_AXIS ... 71
9.3.5 KIN_TARGET_CART ... 72
9.3.6 TARGET-Überschleifen ... 73
9.4 ContinuousPath-Betriebsarten ... 74
9.4.1 Prinzip ... 74
9.4.2 KIN_LIN_XY / _YZ / _ZX ... 75
9.4.3 KIN_LIN_3D ... 75
9.4.4 KIN_CIRC_XY / _YZ / _ZX ... 75
9.4.5 ContinuousPath-Überschleifen ... 76
9.5 Vergleich von ContinuousPath- und TARGET- Überschleifen ... 78
9.6 Kombination von ContinuousPath- und TARGET-Betriebsarten ... 79

10 MOVI-PLC®-Programm ... 81
10.1 Taskkonfiguration ... 81
10.2 Anwenderprogramm ... 82
10.3 Beispielprogramme ... 83
10.3.1 TARGET-Programmierung ... 84
10.3.2 ContinuousPath-Programmierung ... 86

11 Variablenschnittstelle AxisGroupKin.Inst[..] ... 88
11.1 In.General ... 90
11.2 In.Homing ... 92
11.3 In.Transform ... 93
11.4 In.MasterPosition ... 94
11.5 In.Jog ... 95
11.6 In.Target ... 95
11.7 In.Cp ... 98
11.7.1 In.Cp.Settings ... 99
11.7.2 In.Cp.BackToPath ... 99
11.7.3 In.Cp.Segment ... 100
11.7.4 In.Cp.Path ... 104
11.8 In.Simu3D ... 105
11.9 In.Diag ... 106
11.10 Out.General ... 107
11.11 Out.Homing ... 112
11.12 Out.Cp ... 112
11.13 Out.Simu3D ... 114
11.14 Out.Diag ... 114

12 Bausteine der Bibliothek MPLCKinematics ... 116
12.1 Funktion MC_KinInProximity ... 116

20211732 / DE – 07/2014

12.2 Funktionsbaustein MC_KinCoordSysMeasurement ... 117
12.3 Funktionsbaustein MC_KinEncoderDataProcessing ... 119
12.4 Funktionsbaustein MC_KinWcsPcs1Assignment ... 122
12.5 Funktionsbaustein MC_KinWcsPcs2Assignment ... 123
12.6 Funktionsbaustein MC_KinProfGen ... 124

13 Anwendungsfälle ... 127
13.1 Referenzieren eines Roller-Gantry ... 127
13.2 Offene Parallelkinematik ... 127
13.3 Wechsel des Koordinatensystems ... 127
13.4 Vermessen eines Koordinatensystems ... 130
13.5 Programmstruktur für Tracking-Anwendungen mit PCS1/2 ... 132
13.6 Programmstruktur für Tracking-Anwendungen mit mehr als 2 PCS ... 133
13.7 Verwendung einer Tool-Transformation ... 135
13.8 Kombination von Kurvenscheiben- und Kinematikfunktionalität ... 136
13.9 CP-Interpolation als Slave eines externen Masterprofils ... 136
13.10 Kontinuierliche Veränderung der Kinematik-Parameter ... 137
13.11 Kinematiken mit mehr als 6 Antrieben ... 137
13.12 Einbinden einer User-Kinematik ... 138
13.13 Umgang mit Mehrdeutigkeiten der ABC-Orientierungswerte ... 138
13.13.1 Eindeutige Zuordnung ... 138
13.13.2 Keine Eindeutigkeit ... 139
13.13.3 Übernahme der abgespeicherten Orientierungswerte ... 139
13.14 Sicherung von Resolver-Positionswerten ... 140

14 Problembehandlung ... 141
14.1 3D-Simulation ... 141
14.1.1 Kein Verbindungsaufbau möglich ... 141
14.1.2 Es erscheint kein Modell ... 142
14.2 Lastvorsteuerung ... 143
14.2.1 Durchsacken der Achsen ... 143
14.3 Parametrierung der Umrichter ... 144

15 Fehlercodes ... 148
15.1 Allgemeine Fehler ... 148
15.2 Konfigurations-Fehler ... 150
15.3 Allgemeine Parameter-Fehler ... 152
15.4 Target Parameter-Fehler ... 153
15.5 ContinuousPath Parameter-Fehler ... 154
15.6 Profil-Generator Fehler ... 157
15.7 3D-Simulation Fehler ... 158
15.8 AxisGroupControl Kinematik Fehler ... 159

16 Glossar und Abkürzungsverzeichnis ... 161
16.1 Glossar ... 161
16.2 Abkürzungsverzeichnis ... 167

Stichwortverzeichnis ... 169

20211732 / DE – 07/2014

1 Allgemeine Hinweise

1.1 Gebrauch der Dokumentation

Die Dokumentation ist Bestandteil des Produkts und enthält wichtige Hinweise. Die Dokumentation wendet sich an alle Personen, die Arbeiten am Produkt ausführen.

Die Dokumentation muss in einem leserlichen Zustand zugänglich gemacht werden. Stellen Sie sicher, dass die Anlagen- und Betriebsverantwortlichen, sowie Personen, die unter eigener Verantwortung mit der Software und den angeschlossenen Geräten von SEW-EURODRIVE arbeiten, die Dokumentation vollständig gelesen und verstanden haben. Bei Unklarheiten oder weiterem Informationsbedarf wenden Sie sich an SEW-EURODRIVE.

1.2 Aufbau der Sicherheitshinweise

1.2.1 Bedeutung der Signalworte

Die folgende Tabelle zeigt die Abstufung und Bedeutung der Signalworte der Warnhinweise.

Signalwort	Bedeutung	Folgen bei Missachtung
⚠ **GEFAHR**	Unmittelbar drohende Gefahr	Tod oder schwere Verletzungen
⚠ **WARNUNG**	Mögliche, gefährliche Situation	Tod oder schwere Verletzungen
⚠ **VORSICHT**	Mögliche, gefährliche Situation	Leichte Verletzungen
ACHTUNG	Mögliche Sachschäden	Beschädigung des Antriebssystems oder seiner Umgebung
HINWEIS	Nützlicher Hinweis oder Tipp: Erleichtert die Handhabung des Antriebssystems.	

1.2.2 Aufbau der abschnittsbezogenen Warnhinweise

Die abschnittsbezogenen Warnhinweise gelten nicht nur für eine spezielle Handlung, sondern für mehrere Handlungen innerhalb eines Themas. Die verwendeten Gefahrensymbole weisen entweder auf eine allgemeine oder spezifische Gefahr hin.

Hier sehen Sie den formalen Aufbau eines abschnittsbezogenen Warnhinweises:

SIGNALWORT!

Art der Gefahr und ihre Quelle.

Mögliche Folge(n) der Missachtung.

- Maßnahme(n) zur Abwendung der Gefahr.

20211732 / DE – 07/2014

Bedeutung der Gefahrensymbole

Die Gefahrensymbole, die in den Warnhinweisen stehen, haben folgende Bedeutung:

Gefahrensymbol	Bedeutung
	Allgemeine Gefahrenstelle
	Warnung vor gefährlicher elektrischer Spannung
	Warnung vor heißen Oberflächen
	Warnung vor Quetschgefahr
	Warnung vor schwebender Last
	Warnung vor automatischem Anlauf

1.2.3 Aufbau der eingebetteten Warnhinweise

Die eingebetteten Warnhinweise sind direkt in die Handlungsanleitung vor dem gefährlichen Handlungsschritt integriert.

Hier sehen Sie den formalen Aufbau eines eingebetteten Warnhinweises:

- ⚠ **SIGNALWORT!** Art der Gefahr und ihre Quelle.

  Mögliche Folge(n) der Missachtung.

  – Maßnahme(n) zur Abwendung der Gefahr.

1.3 Mängelhaftungsansprüche

Die Einhaltung der vorliegenden Dokumentation ist die Voraussetzung für einen störungsfreien Betrieb und die Erfüllung eventueller Mängelhaftungsansprüche. Lesen Sie deshalb zuerst die Dokumentationen, bevor Sie mit der Software und den angeschlossenen Geräten von SEW-EURODRIVE arbeiten!

Stellen Sie sicher, dass die Dokumentationen den Anlagen- und Betriebsverantwortlichen, sowie Personen, die unter eigener Verantwortung an den Geräten arbeiten, in einem leserlichen Zustand zugänglich gemacht werden.

20211732 / DE – 07/2014

1.4 Haftungsausschluss

Beachten Sie die vorliegende Dokumentation und die mitgeltende Dokumentation zur verwendeten Software sowie den angeschlossenden Geräten von SEW-EURODRIVE. Dies ist die Grundvoraussetzung, um einen sicheren Betrieb, die angegebenen Produkteigenschaften und Leistungsmerkmale zu erreichen.

SEW-EURODRIVE übernimmt für Personen-, Sach- oder Vermögensschäden, die wegen Nichtbeachtung der Dokumentation entstehen, keine Haftung. SEW-EURODRIVE schließt die Sachmängelhaftung in solchen Fällen aus.

1.5 Mitgeltende Unterlagen

Beachten Sie folgende mitgeltenden Unterlagen:

- Betriebsanleitung "Mehrachs-Servoverstärker MOVIAXIS® MX"
- Betriebsanleitung "Antriebsumrichter MOVIDRIVE® MDX"
- Betriebsanleitung "Frequenzumrichter MOVITRAC® MC07"
- Handbuch "Controller DH.21B (standard) / DH.41B (advanced)"
- Handbuch "Controller UHX71B (power)"
- Handbuch "Programm-Modul MultiMotion, universelle, parametrierbare Software-Plattform für MOVI-PLC®"
- Systemhandbuch "MOVI-PLC®-Programmierung im PLC-Editor"
- Handbuch „Technologie-Modul HandlingKinematics für MultiMotion / MultiMotion Light“

Verwenden Sie immer die aktuelle Ausgabe der Dokumentation und Software.

Auf der Homepage von SEW-EURODRIVE (www.sew-eurodrive.de) finden Sie eine große Auswahl von Dokumentationen in verschiedenen Sprachen zum Herunterladen.

Bei Bedarf können Sie die Druckschriften in gedruckter und gebundener Form bei SEW-EURODRIVE bestellen.

1.6 Produktnamen und Marken

Die in dieser Dokumentation genannten Produktnamen sind Marken oder eingetragene Marken der jeweiligen Titelhalter.

1.7 Urheberrechtsvermerk

© 2014 SEW-EURODRIVE. Alle Rechte vorbehalten.

Jegliche – auch auszugsweise – Vervielfältigung, Bearbeitung, Verbreitung und sonstige Verwertung sind verboten.

20211732 / DE – 07/2014

2 Sicherheitshinweise

2.1 Allgemein

Die folgenden grundsätzlichen Sicherheitshinweise dienen dazu, Personen- und Sachschäden zu vermeiden. Der Betreiber muss sicherstellen, dass die grundsätzlichen Sicherheitshinweise beachtet und eingehalten werden.

Vergewissern Sie sich, dass Anlagen- und Betriebsverantwortliche sowie Personen, die unter eigener Verantwortung arbeiten, die Dokumentationen vollständig gelesen und verstanden haben. Bei Unklarheiten oder weiterem Informationsbedarf wenden Sie sich an SEW-EURODRIVE.

Die folgenden Sicherheitshinweise beziehen sich auf den Einsatz der Software. Berücksichtigen Sie auch die ergänzenden Sicherheitshinweise in dieser Dokumentation und in den Dokumentationen zu den angeschlossenen Geräten von SEW-EURODRIVE.

Diese Dokumentation ersetzt nicht die ausführlichen Dokumentationen der angeschlossenen Geräte! Die vorliegende Dokumentation setzt das Vorhandensein und die Kenntnis der Dokumentationen zu allen angeschlossenen Geräten von SEW-EURODRIVE voraus.

Niemals beschädigte Produkte installieren oder in Betrieb nehmen. Beschädigungen umgehend beim Transportunternehmen reklamieren.

Während des Betriebs können die Geräte ihrer Schutzart entsprechend spannungsführende, blanke gegebenenfalls auch bewegliche oder rotierende Teile sowie heiße Oberflächen haben.

Bei unzulässigem Entfernen der erforderlichen Abdeckung, unsachgemäßem Einsatz, bei falscher Installation oder Bedienung, besteht die Gefahr von schweren Personen- oder Sachschäden. Weitere Informationen sind der Dokumentation zu entnehmen.

2.2 Zielgruppe

Alle Arbeiten mit der eingesetzten Software dürfen ausschließlich von einer ausgebildeten Fachkraft ausgeführt werden. Fachkraft im Sinne dieser Dokumentation sind Personen, die über folgende Qualifikationen verfügen:

- Geeignete Unterweisung.
- Kenntnis dieser Dokumentation und der mitgeltenden Dokumentationen.
- SEW-EURODRIVE empfiehlt zusätzlich Produktschulungen zu den Produkten, die mit dieser Software betrieben werden.

Alle mechanischen Arbeiten an den angeschlossenen Geräten dürfen ausschließlich von einer ausgebildeten Fachkraft ausgeführt werden. Fachkraft im Sinne dieser Dokumentation sind Personen, die mit Aufbau, mechanischer Installation, Störungsbehebung und Instandhaltung des Produkts vertraut sind und über folgende Qualifikationen verfügen:

- Ausbildung im Bereich Mechanik (beispielsweise als Mechaniker oder Mechatroniker) mit bestandener Abschlussprüfung.
- Kenntnis dieser Dokumentation und der mitgeltenden Dokumentationen.

Alle elektrotechnischen Arbeiten an den angeschlossenen Geräten dürfen ausschließlich von einer ausgebildeten Elektrofachkraft ausgeführt werden. Elektrofachkraft im Sinne dieser Dokumentation sind Personen, die mit elektrischer Installation, Inbetriebnahme, Störungsbehebung und Instandhaltung des Produkts vertraut sind und über folgende Qualifikationen verfügen:

20211732 / DE – 07/2014

- Ausbildung im Bereich Elektrotechnik (beispielsweise Elektroniker oder Mechatroniker) mit bestandener Abschlussprüfung.
- Kenntnis dieser Dokumentation und der mitgeltenden Dokumentationen.
- Kenntnis der jeweils gültigen Sicherheitsvorschriften und Gesetze.
- Kenntnis der anderen in dieser Dokumentation genannten Normen, Richtlinien und Gesetze.

Die genannten Personen müssen die betrieblich ausdrücklich erteilte Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu programmieren, zu parametrieren, zu kennzeichnen und zu erden.

Alle Arbeiten in den übrigen Bereichen Transport, Lagerung, Betrieb und Entsorgung dürfen ausschließlich von Personen durchgeführt werden, die in geeigneter Weise unterwiesen wurden.

2.3 Bestimmungsgemäße Verwendung

Das Programm-Modul MultiMotion mit dem Technologiemodul "Kinematics" ist eine universelle, parametrierbare Software-Plattform für die Controller MOVI-PLC® advanced und power von SEW-EURODRIVE.

Es besteht aus folgenden Komponenten:

- **Projektvorlage "AxisControl_MultiMotion.pro"**

  Der Anwender erstellt aus der Projektvorlage ein MOVI-PLC®-Projekt. Anschließend importiert er das Technologiemodul "Kinematics" und die dazu passende Task-Konfiguration. Damit sind alle Voraussetzungen geschaffen, dass er im MOVI-PLC®-Programm auf die Kinematikfunktionen zugreifen kann.

- **MultiMotion Editor**

  Das Werkzeug dient zum einen der Parametrierung der im MOVI-PLC®-Programm enthaltenen Funktionen und zum anderen der Diagnose und dem Test der parametrierten Funktionen.

Das Programm-Modul MultiMotion mit dem Technologiemodul "Kinematics" stellt keine fertige Anwendungslösung dar, sondern eine Software-Vorlage, die der Anwender um eigene Programmteile ergänzen muss. Ohne diese Ergänzungen ist MultiMotion nicht geeignet, automatische Abläufe in Maschinen zu steuern.

Test im Steuerbetrieb

Beim Testen der parametrierten Funktionen mit dem MultiMotion Editor kann im Steuerbetrieb direkt auf die Schnittstelle und damit unmittelbar auf die Antriebsfunktionen zugegriffen werden. D. h. die Schnittstelle wird nicht mehr durch die vom Anwender ergänzten Programmteile gesteuert, wodurch die Begrenzungen und Verriegelungen im Anwenderprogramm unwirksam sein können. Daher sind bei der Nutzung des Steuerbetriebs entsprechende Vorsichtsmaßnahmen zu treffen. Die Nutzung des Steuerbetriebs liegt in der alleinigen Verantwortung des Anwenders.

2.4 Bussysteme

Mit einem Bussystem ist es möglich, Frequenzumrichter und/oder Motorstarter in weiten Grenzen an die Anlagengegebenheiten anzupassen. Dadurch besteht die Gefahr, dass die von außen nicht sichtbare Änderung der Parameter zu einem unerwarteten, aber nicht unkontrollierten Systemverhalten führen kann.

20211732 / DE – 07/2014

3 Systembeschreibung

3.1 Inhalt dieses Handbuchs

Das Handbuch erläutert die Funktionalität, Inbetriebnahme und Bedienung der Kinematiksteuerung des Technologiemoduls "Kinematics für MultiMotion / MultiMotion Light".

Im Folgenden erhalten Sie einen Überblick darüber, was Sie in den einzelnen Kapitel erwartet:

Grundlagen der Robotik	Geht zunächst auf grundlegende Aspekte der Kinematiksteuerungen ein, die später detailliert behandelt werden.
Auswahl von Kinematics oder Handling-Kinematics	Es stehen 2 verschiedene Technologiemodule zur Steuerung von Kinematiken zur Verfügung. Dieses Kapitel stellt Informationen zur Auswahl zur Verfügung.
Inbetriebnahme und Konfiguration	Erläutert die ersten Schritte zur Erstellung des Projekts, die Konfiguration sowie die Inbetriebnahme. Außerdem wird in diesem Kapitel die grundlegende Bedienung mithilfe des Kinematikmonitors erläutert.
Kinematikmodelle	Gibt einen Überblick über die konfigurierbaren Kinematikmodelle. (Detaillierte Informationen entnehmen Sie dem Konfigurationsassistenten im MultiMotion-Editor.)
Interpolierende Betriebsarten	Erläutert die zur Verfügung stehenden Betriebsarten zur Bahninterpolation und zeigt die Eignung für verschiedene Anwendungsbereiche.
MOVI-PLC®-Programm	Erläutert die Struktur des MOVI-PLC®-Programms und zeigt Beispiele von Programmteilen, die zur Steuerung automatischer Abläufe in Maschinen dienen.
Variablenschnittstelle *AxisGroup-Kin.Inst[..]*	Erläutert detailliert die globale Variablenschnittstelle zur Steuerung der beschriebenen Kinematikfunktionen.
Bausteine der Bibliothek MPLCKinematics	Hier werden Bausteine vorgestellt, die sich für die Verwendung im Anwenderprogramm anbieten.
Anwendungsfälle	Beschreibt die Nutzung spezieller Kinematikfunktionen. Referenziert auf Variablen und Bausteine aus den vorherigen Kapiteln.
Problembehandlung	Zeigt mögliche Probleme und gibt Hinweise zur Abhilfe.
Fehlercodes	Listet die Fehlercodes auf
Glossar und Abkürzungsverzeichnis	Erläutert die im Handbuch verwendeten Fachbegriffe und Abkürzungen.

20211732 / DE – 07/2014

3.1.1 Schreibkonventionen

- Das Handbuch enthält sehr viele Objekte (Pfade, Variablen, Bibliotheken etc.), die sich zum besseren Verständnis von der Schreibweise unterscheiden. Die folgende Tabelle zeigt die Schreibkonvention, die dafür angewendet wird:

Objekt	Schreibweise	Beispiel
Pfad	Courier-Schrift	`\TechModules\Kinematics`
Eingangssignal (Variable) Ausgangssignal	kursiv	*AxisGroupKin.Inst[..].In.General.Ignore.SWLS.Cart*
Betriebsart	fett, durchgehende Großschreibung	**KIN_JOG_CART**
Programm, Programmabschnitt	fett, kursiv	**PRG_TaskMain**

- Für einige hier verwendeten Fachausdrücke existieren englischsprachige Entsprechungen und geläufige (ebenfalls englischsprachige) Abkürzungen. In dem Handbuch finden Sie die deutschsprachige Bezeichnung und die englischsprachige Abkürzung in Klammern wie z. B. Werkstückkoordinatensystem (PCS). Details entnehmen Sie dem Kapitel "Glossar und Abkürzungsverzeichnis" (→ 🗎 161).

3.2 Voraussetzungen

3.2.1 Hardware

Für den Einsatz des Technologiemoduls "Kinematics für MultiMotion / MultiMotion Light" gelten die folgenden Hardware-Voraussetzungen:

- MOVI-PLC® advanced oder MOVI-PLC® power
- SD/CFast-Speicherkarte mit ausreichend hoher Technologiestufe.

  Die benötigte Technologiestufe hängt von der gewünschten Funktionalität ab (siehe Kapitel "Erforderliche Technologiestufe" (→ 🗎 13)).

3.2.2 Software

Es gelten die folgenden Software-Voraussetzungen:

- Programm-Modul MultiMotion oder MultiMotion Light V150r100 oder höher
- Technologie-Modul Kinematics V150r100 oder höher

Beide Softwarekomponenten sind in MOVITOOLS® MotionStudio 6.0 oder höher automatisch enthalten.

Auf der Homepage von SEW-EURODRIVE (www.sew-eurodrive.de) finden Sie die Software zum Herunterladen.

3.2.3 Erforderliche Technologiestufe

Die Technologiestufe ergibt sich aus den für die eingesetzten Funktionalitäten erforderlichen Technologiepunkten. Sie können sie entweder mit den folgenden Informationen errechnen. Oder Sie parametrieren die gewünschte Funktionalität im MultiMotion-Konfigurationsassistent, der die erforderlichen Technologiepunkte anzeigt.

Für die Berechnung werden folgende Technologiepunkte berechnet:

20211732 / DE – 07/2014

- Technnologiepunkte für die Betriebsart (ggf. mit zusätzl. Betriebsart)
- Zusätzliche Technologiepunkte für die Funktionen

Die folgende Tabelle zeigt wieviel Technologiepunkte für die jeweilige Betriebsart (pro Kinematikinstanz) verbraucht werden:

Betriebsart	Technologiepunkte (pro Kinematikinstanz)
AM_HOMING KIN_JOG_AXIS KIN_JOG_CART KIN_TARGET_AXIS KIN_TARGET_CART	2
AM_HOMING KIN_JOG_AXIS KIN_JOG_CART KIN_TARGET_AXIS KIN_TARGET_CART KIN_LIN_XY/YZ/ZX KIN_CIRC_XY/YZ/ZX	3
AM_HOMING KIN_JOG_AXIS KIN_JOG_CART KIN_TARGET_AXIS KIN_TARGET_CART KIN_LIN_XY/YZ/ZX KIN_CIRC_XY/YZ/ZX KIN_LIN_3D	4

Die folgende Tabelle zeigt wieviel Technologiepunkte für die jeweilige Funktion (pro Kinematikinstanz) verbraucht werden:

Funktion	zusätzl. Technologie-punkte (pro Kinematikinstanz)
Kinematikmodell ist kein CARTESIAN_GANTRY	+1
Verwendung des Welt- oder eines Werkstückkoordinaten-systems (WCS, PCS1/2), z. B. für Tracking Anwendungen oder Palettieren mit variierenden Palettenpositionen.	+1

20211732 / DE – 07/2014

Die folgende Tabelle zeigt wieviel Technologiepunkte für die jeweilige Funktion (pro MOVI-PLC®) verbraucht werden:

Funktion	zusätzl. Technologiepunkte (pro MOVI-PLC®)
3D-Simulation: • Wird gewöhnlich nur für MOVI-PLC benötigt, auf der Bewegungs- und Ablaufprogramme erstellt werden. Hierfür steht Ihnen eine Speicherkarte mit einer Einzelplatzlizenz für Entwickler (z. B. mit 18 Technologiepunkten) zur Verfügung.	+10[1)]
MultiMotion "Full": • Für das Technologiemodul "Kinematics" ist die Ausprägung "Full" NICHT erforderlich. • MultiMotion "Full" ist nur erforderlich, wenn darüber hinausgehend MultiMotion-Achsen genutzt werden, für die MultiMotion Light nicht ausreicht.	+2

1) 30 Min. Testbetrieb ohne zusätzliche Technologiepunkte möglich

Für die Nutzung der Technologiepunkte ist Folgendes zu beachten:

- Auf der Speicherkarte der MOVI-PLC® muss die erforderliche Anzahl von Technologiepunkten vorhanden sein, sonst kann die Funktionalität nicht genutzt werden.
- Die erforderlichen Technologiepunkte werden beim Start der MOVI-PLC® verbraucht, unabhängig davon, ob die Funktionalität tatsächlich im MOVI-PLC®-Projekt genutzt wird.

Berechnungsbeispiel

Die Tabelle zeigt die Berechnung der Technologiepunkte für eine Palettieranwendung, realisiert mit einem SCARA-Roboter mit LIN_3D_Bahninterpolation:

Funktionalität		Technologiepunkte
Betriebsart	LIN_3D	4
Funktionen	SCARA (Kinematikmodell ist kein CARTESIAN_GANTRY)	1
	Verwendung eines Werkstückkoordinatensystems (PCS)	1
Summe		6

3.3 Funktionen

Die umfassenden Funktionen der universell parametrierbaren Software-Plattform MultiMotion sind im Handbuch "Programm-Modul MultiMotion: Universelle, parametrierbare Software-Plattform für MOVI-PLC®" beschrieben und in vollem Umfang nutzbar.

Das Technologiemodul "Kinematics" verwendet pro konfigurierte Kinematikinstanz 8 MultiMotion-Achsen und kommuniziert mit diesen über die globale MultiMotion-Variablenschnittstelle "AxisInterface".

20211732 / DE – 07/2014

Das Technologiemodul bietet folgende Funktionalität:

- Tippbetrieb der Einzelachsen (**KIN_JOG_AXIS**)
- Tippbetrieb der kartesischen TCP-Koordinaten (**KIN_JOG_CART**)
- Positionieren der Einzelachsen, nicht synchronisiert oder synchronisiert (**KIN_TARGET_AXIS**)
- Positionieren der kartesischen TCP-Koordinaten, nicht synchronisiert oder synchronisiert (**KIN_TARGET_CART**)
- Bewegung entlang geometrisch definierten Geraden- und Kreissegmenten mit automatischem Überschleifen (ContinuousPath, z. B. **KIN_LIN_..**, **KIN_CIRC**)
- Steuerung in verschiedenen, beweglichen Koordinatensystemen (Kinematik, Welt, Werkstücke)
- Werkzeugtransformation (*Transform.Tool*)
- Master-Slave-Beziehungen (z. B. Abfahren einer zentral gerechneten Kurve mithilfe einer Kinematik)
- Möglichkeit der Implementierung von Kinematikmodellen für spezielle Anwendungen (USER-Kinematik)
- Steuerung von bis zu 8 Freiheitsgraden (DOF), jedoch besteht zusätzlich die Möglichkeit der Integration weiterer Achsen in die Kinematik, z. B. von außen gesteuerte Teleskopachse oder Achse zum Verfahren der gesamten Kinematik. u. v. m.

20211732 / DE – 07/2014

3.4 Anwendungsbereiche

Das Technologiemodul "Kinematics" ist aufgrund der vielfältigen Betriebsarten und Schnittstellen zur Steuerung jeder vorstellbaren offenen und geschlossenen kinematischen Kette in verschiedenen Anwendungsbereichen geeignet.

- Handling
- Palettieren / Depalettieren
- Tracking
- Dekoration (z. B. von Teigwaren)
- Montage
- Bearbeitung, (z. B. Fräsen, Kleben, Lackieren)
- Kameraführung
- u. v. m.

Einen Schwerpunkt bildet die Handhabung von Objekten. Die zur Verfügung stehenden Betriebsarten zur Bahninterpolation eignen sich insbesondere für Handling-Aufgaben wie Pick-and-Place, Palettieren oder Tracking.

20211732 / DE – 07/2014

4 Grundlagen der Robotik

4.1 Bahnsteuerung

Im Gegensatz zur Einzelachssteuerung wird bei der Bahnsteuerung die Bewegung eines Werkzeugs im 3D-Raum gesteuert. Der Anwender gibt die Zielposition, die Form der Bahn (z. B. eine Gerade oder einen Kreisbogen) sowie die kinematischen Bewegungsgrößen (z. B. Geschwindigkeit, Beschleunigung) an. Die Kinematiksteuerung verfährt alle Achsen automatisch in geeigneter Weise, so dass sich das Werkzeug entlang der gewünschten Bahn zur Zielposition bewegt.

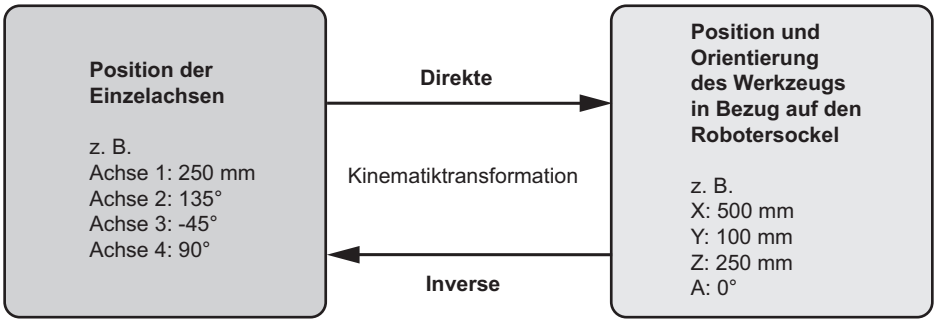
Die Kinematiksteuerung unterstützt neben der Bahnsteuerung auch die herkömmliche Steuerung der Achsen und kann fließend zwischen den verschiedenen Betriebsarten umschalten.

4.2 Arten der Kinematiktransformation

Direkte Transformation: Die direkte Kinematiktransformation berechnet aus den Positionen der Einzelachsen Position und Orientierung des Roboterwerkzeugs in Bezug auf den Robotersockel. Diese Transformation wird beispielsweise bei der Aktivierung der Kinematiksteuerung ausgeführt, um aus den Motorinkrementen die kartesischen XYZ-Koordinaten zu bestimmen.

Inverse Kinematiktransformation: Die inverse Kinematiktransformation berechnet die Positionen der Achsen, mit denen das Werkzeug des Roboters eine bestimmte, kartesische Raumposition erreicht. Diese Transformation wird beispielsweise während der Bahninterpolation ausgeführt, um die Einzelachsen in geeigneter Weise führen zu können, so dass sich das Werkzeug entlang der gewünschten Bahn bewegt.

Die folgende Darstellung zeigt die Verwendung der direkten und der inversen Kinematiktransformation:



9290210443

4.3 Beispiele für weitere Transformationen

Die kartesische Werkzeugposition des Roboters lässt sich nicht nur in Bezug auf den Robotersockel angeben, sondern auch z. B. in Bezug auf ein frei wählbares Welt- (WCS) oder Werkstückkoordinatensystem (PCS).

Ein Werkstückkoordinatensystem (PCS) kann z. B. in der Ecke eine Palette platziert sein. Die Positionen der Kartons auf der Palette können dann in Bezug auf dieses Werkstückkoordinatensystem (PCS) angegeben werden, z. B. X = 200 mm , Y = 300 mm. Der Palettier-Roboter steht möglicherweise 2 m von der Palette entfernt. Und zusätzlich ist die Palette eventuell gegenüber dem Robotersockel verdreht.

20211732 / DE – 07/2014

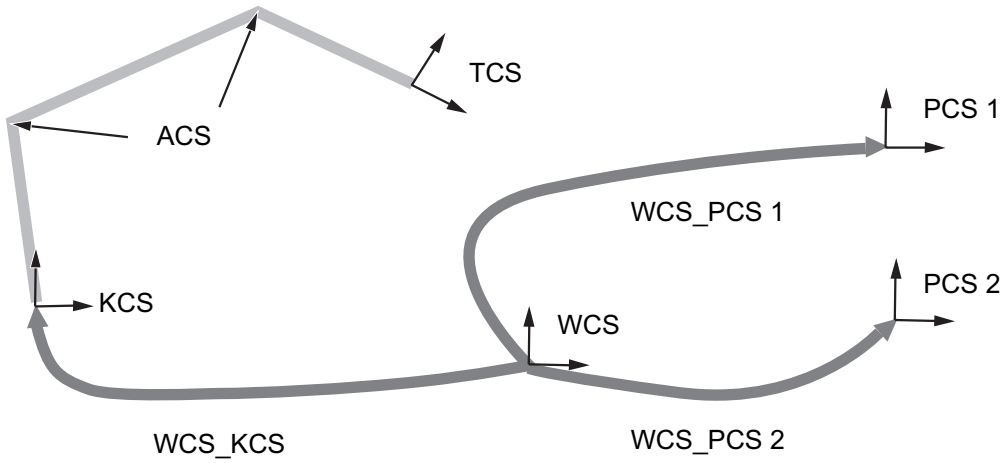
Wenn in diesem Beispiel die Transformation vom Robotersockel zum Werkstückkoordinatensystem (PCS) in die Kinematiksteuerung eingespeist wird, kann der Roboter einfach im Werkstückkoordinatensystem (PCS) zu den Kartonkoordinaten X = 200 mm, Y = 300 mm verfahren werden. Der Anwender muss also nicht mühevoll berechnen, welche Position die Kartons in Bezug auf den Robotersockel hätten.

Die hierzu erforderliche Transformation besteht einfach in der Angabe der Position und Verdrehung des Werkstückkoordinatensystems (PCS) in Bezug auf den Robotersockel. Wenn die Palette bei jedem Palettiervorgang z. B. etwas verrutscht oder verdreht daliegt, müssen nur diese Versätze erfasst und der Kinematiksteuerung übermittelt werden. Die Koordinaten des Palettiermusters bleiben unverändert.

Wenn sich ein Werkstückkoordinatensystem (PCS) in Bewegung befindet, z. B. weil ein mit Zuckerguss zu verzierender Kuchen auf einem Transportband am Roboter vorbeigeführt wird, können so auf einfache Art und weise Verzierungen wie z. B. Kreise, Beschriftungen u. ä. an bewegten Objekten erzeugt werden.

Auch der Roboter selbst kann durch eine oder mehrere zusätzliche Achsen bewegt werden, z. B. um seinen Arbeitsraum zu vergrößern. Auch die hierbei veränderliche Transformation von einem Weltkoordinatensystem , z. B. Bezugspunkt in einer Halle, zum bewegten Robotersockel lässt sich einfach in die Kinematiksteuerung einspeisen, so dass die Bewegung durch die Roboterachsen automatisch kompensiert wird.

Die Werkzeugtransformation ist eine weitere, häufig genutzte Möglichkeit zur Erweiterung der kinematischen Kette. Die direkte Kinematiktransformation endet am Roboterflansch. Sollen sich die kartesischen Koordinaten nicht auf den Flansch, sondern z. B. auf den Mittelpunkt zwischen den Backen eines Greifers beziehen, wird die Position dieses Tool-Center-Points in Bezug auf den Flansch in die Kinematiksteuerung eingespeist.



12549710731

4.4 Kartesische Position

Die kartesische Pose (Position und Orientierung) des Werkzeugkoordinatensystems (TCS) in einem bestimmten kartesischen Koordinatensystem, z. B. KCS, wird mithilfe folgender Koordinaten ausgedrückt:

- Positionskoordinaten: X, Y und Z
- Orientierungskoordinaten: A, B und C

Die Positionskoordinaten X, Y, Z geben die Position des Ursprungs des Werkzeugkoordinatensystems (TCP) im Bezugskoordinatensystem an, z. B. KCS.

20211732 / DE – 07/2014

Die Orientierungskoordinaten A, B, C geben an, wie das Werkzeugkoordinatensystem (TCS) gegenüber dem Bezugskoordinatensystem verdreht ist. Sind alle 3 Werte gleich 0, haben das Werkzeugkoordinatensystem (TCS) und das Bezugskoordinatensystem die gleiche Orientierung.

Ansonsten wird die Orientierung des Werkzeugkoordinatensystems (TCS) aus der Orientierung des Bezugskoordinatensystems folgendermaßen abgeleitet:

1. Verdrehung des Bezugskoordinatensystems mit dem Wert A um seine Z-Achse
2. Verdrehung des im ersten Schritt verdrehten Koordinatensystems mit dem Wert B um seine Y-Achse, also um die neue Y-Achse
3. Verdrehung des im zweiten Schritt verdrehten Koordinatensystems mit dem Wert C um seine X-Achse, also um die neue X-Achse

In der gleichen Weise werden jeweils mithilfe eines 6-dimensionalen Vektors auch die folgenden Transformationen beschrieben:

- Werkzeugtransformation:

  Lage des Werkzeugkoordinatensystems (TCS) relativ zum Flanschkoordinatensystem (FCS)
- Transformation vom Weltkoordinatensystem (WCS) zum Kinematikkoordinatensystem (KCS)
- Transformation vom Weltkoordinatensystem (WCS) zu einem Werkstückkoordinatensystem (PCS)

20211732 / DE – 07/2014

5 Auswahl von Kinematics oder HandlingKinematics

5.1 Ebene 1: Applikationsmodul HandlingKinematics für CCU

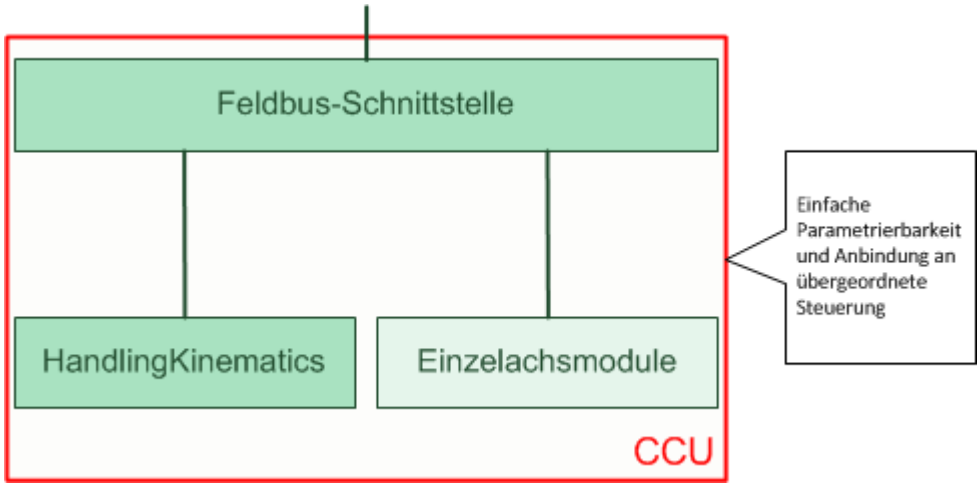
5.1.1 Eigenschaften

- Keine Programmierung
- Die CCU übernimmt alle Antriebsbetriebsarten, die zentrale Steuerung koordiniert den Prozessablauf und definiert die Produktdaten
- Fertige Schnittstelle zur übergeordneten Steuerung mit bis zu 20 Bahnpunkten
- Simulation der Abläufe und Diagnose sogar ohne reale Maschine möglich
- Wählen Sie aus 4 Bewegungsarten das für Ihren Fall perfekte Bewegungsprofil
- Wählen Sie eine Mechanik mit bis 4 Freiheitsgraden XYZ und Drehung um Z
- Reproduzierbare Bahntreue auch nach Störungen mit BACK-TO-PATH
- Geeignet für ruhende Objekte und kombinierbar mit bis zu 8 weiteren Applikationsmodulen, z. B. für Förderbänder, Hubachsen und Greifer
- Wartepunkte bei jedem Bahnsegment definierbar
- Touchprobe-Messfunktion und Restwegpositionierung

5.1.2 Vorteile

- Funktionsgarantie durch vielfach bewährte, gekapselte und einsatzerprobte Bahnsteuerung.
- Extrem schnelle und einfache Inbetriebnahme einer kompletten Kinematik durch intuitiv bedienbare, grafische Software mit übersichtlicher Diagnose- und Monitorfunktion.
- Die Taktzeit wird deutlich reduziert durch taktsynchrone Bahnsteuerung mit Look-Ahead und konturtreue Umfahrung von Störkanten.

Die folgende Darstellung zeigt den prinzipiellen Aufbau des Applikationsmoduls:



12572940299

5.2 Ebene 2: Technologiemodul HandlingKinematics für MOVI-PLC®

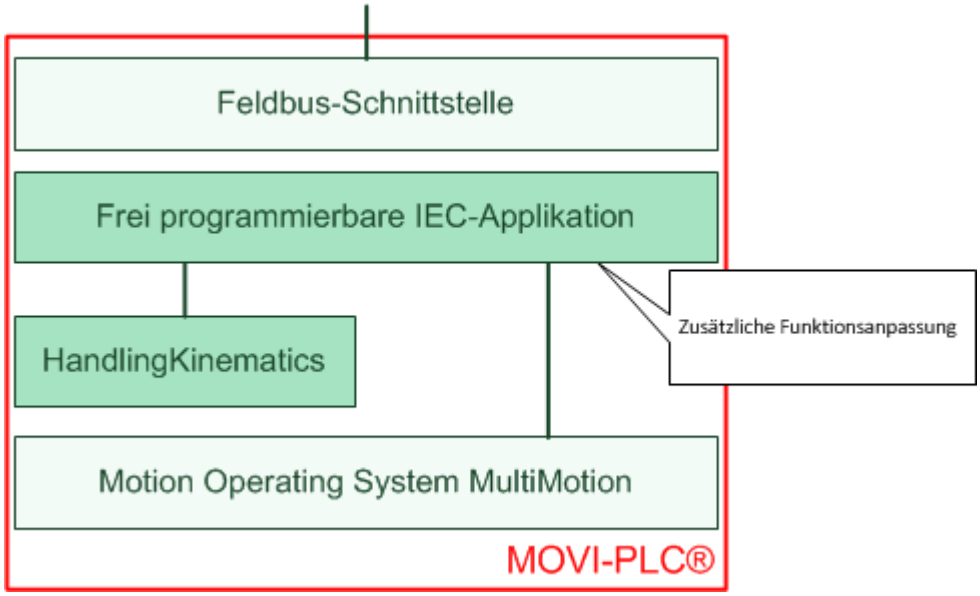
5.2.1 Eigenschaften

- Beinhaltet alle Funktionen des CCU Applikationsmoduls HandlingKinematics.
- Interpretieren und Skalieren Sie die Auflösung der Bahnpunkte nach Ihren Bedürfnissen.
- Praktisch (fast) jeder Roboter ist betreibbar: Wählen Sie eine Mechanik mit bis 6 Freiheitsgraden XYZABC
- Ergänzen und modifizieren Sie die Feldbus-Schnittstelle für Ihre Bedürfnisse: Steuern Sie das Technologiemodul direkt oder über Signale im Ihrem MOVI-PLC®-Programm.

5.2.2 Zusätzliche Vorteile

- Effizienz durch modulares Maschinendesign. Verlagern Sie alle Sensoren und Aktoren, die für ihre Bewegungssteuerung relevant sind, in unsere MOVI-PLC®. Lichtschranken, Näherungsschalter, Vision-Systeme, Pneumatikachsen werden direkt von uns koordiniert.
- Datenaustausch bei Laufzeit auf das Notwendige reduziert: Legen Sie die Rezeptdaten für die Bahnplanung vorab auf der MOVI-PLC® ab.
- Überwachen Sie applikativ die Interaktion mehrerer Kinematiken und insgesamt 64 Antriebe mit unserer Steuerung MOVI-PLC® power
- Eine Software für alles: Betreiben Sie die Hauptmaschine (z. B. Verpackungsmaschine) und das Kinematikmodul mit MultiMotion auf ein und derselben Steuerung.

Die folgende Darstellung zeigt den prinzipiellen Aufbau des Technologiemoduls:



12572944395

20211732 / DE – 07/2014

5.3 Ebene 3: Technologiemodul Kinematics für MOVI-PLC®

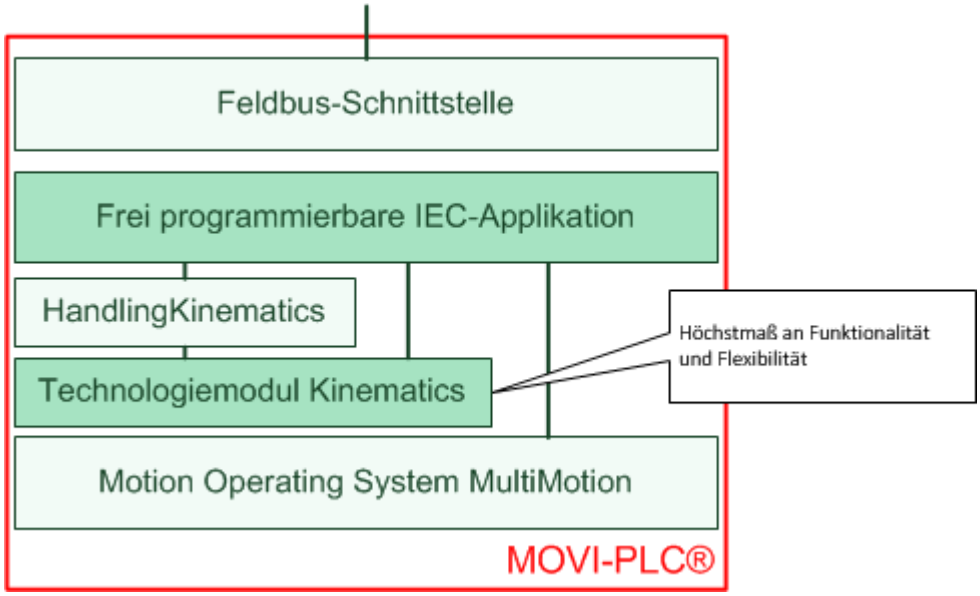
5.3.1 Eigenschaften

- Feingranularer Zugriff auf alle Bewegungsparameter in jedem Bahnsegment
- Umfangreiche Möglichkeiten zur Eingabe von Kreissegmenten
- Alle Koordinatensysteme zur Steuerung nutzbar (auch Achsen-/Welt-/Werkstückkoordinatensystem) und während der Bewegung umschaltbar.
- Mehrere Kinematikinstanzen synchronisierbar auf gemeinsames Werkstück
- Technologiemodule HandlingKinematics und Kinematics auf einem Controller betreibbar.
- Master-Slave-Beziehungen realisierbar, z. B. Bewegung der Kinematik entlang CAM-Profil oder Bahnfortschritt als Funktion von MultiMotion-Achse
- Basic-G-Code-Import für Bewegungsführung entlang CAD-Kontur

5.3.2 Zusätzliche Vorteile

- Lösen Sie einfach und schnell komplexe Aufgaben: Wenn Sie z.B. das gesamte Palettiermuster abhängig von der vorderen Ecke und der Drehung der aktuell eingesetzten Palette mit verschiedenen Koordinatensystemen verschieben wollen, dann erledigt unsere MOVI-PLC® dies in einer Programmzeile in Echtzeit.
- Warten Sie nicht solange, bis Objekte stehen, sondern greifen Sie diese bei voller Fahrt ab (TRACKING) z. B. Werkstückposition als Funktion des Signals eines externen Gebers.
- Eine Software für alles: Betreiben Sie die Hauptmaschine (z. B. Verpackungsmaschine) und das Kinematikmodul mit MultiMotion auf ein und derselben Steuerung.

Die folgende Darstellung zeigt den prinzipiellen Aufbau des Technologiemoduls:



12572948491

20211732 / DE – 07/2014

Hinweis:

Wenn auf einem Controller Instanzen beider Technologiemodule betrieben werden sollen, müssen Sie bei der Erstellung des MOVI-PLC®-Projekts im Schritt 3: Integration des Technologiemoduls (→ 🗎 28) die Bausteine aus dem Verzeichnis „HandlingKinematics“ importieren (statt „Kinematics“). Denn zu jedem Technologiemodul „HandlingKinematics“ existiert unterlagert ein „Technologiemodul“ Kinematics mit 1:1-Zuordnung (HandlingKinematics Instanz n verwendet Kinematics Instanz n).

Beispiel:

Soll z. B. Instanz 2 als Technologiemodul Kinematics betrieben werden, müssen Sie bei der Konfiguration von Instanz 2 „Kinematics“ auswählen (siehe Schritt 2: Erstellung eines MOVI-PLC®-Projekts (→ 🗎 27)). Die HandlingKinematics Bausteine der zweiten Instanz bleiben in dem Fall ungenutzt. Ist jedoch z. B. Instanz 1 als „HandlingKinematics“ konfiguriert (siehe Schritt 2: Erstellung eines MOVI-PLC®-Projekts (→ 🗎 27)), dann verwenden Sie die HandlingKinematics Bausteine und Variablen von Instanz 1. Das unterlagerte Technologiemodul Kinematics von Instanz 1 ist in dem Fall exklusiv für das überlagerte Technologiemodul HandlingKinematics reserviert und wird durch dieses bedient.

20211732 / DE – 07/2014

6 Inbetriebnahme und Konfiguration

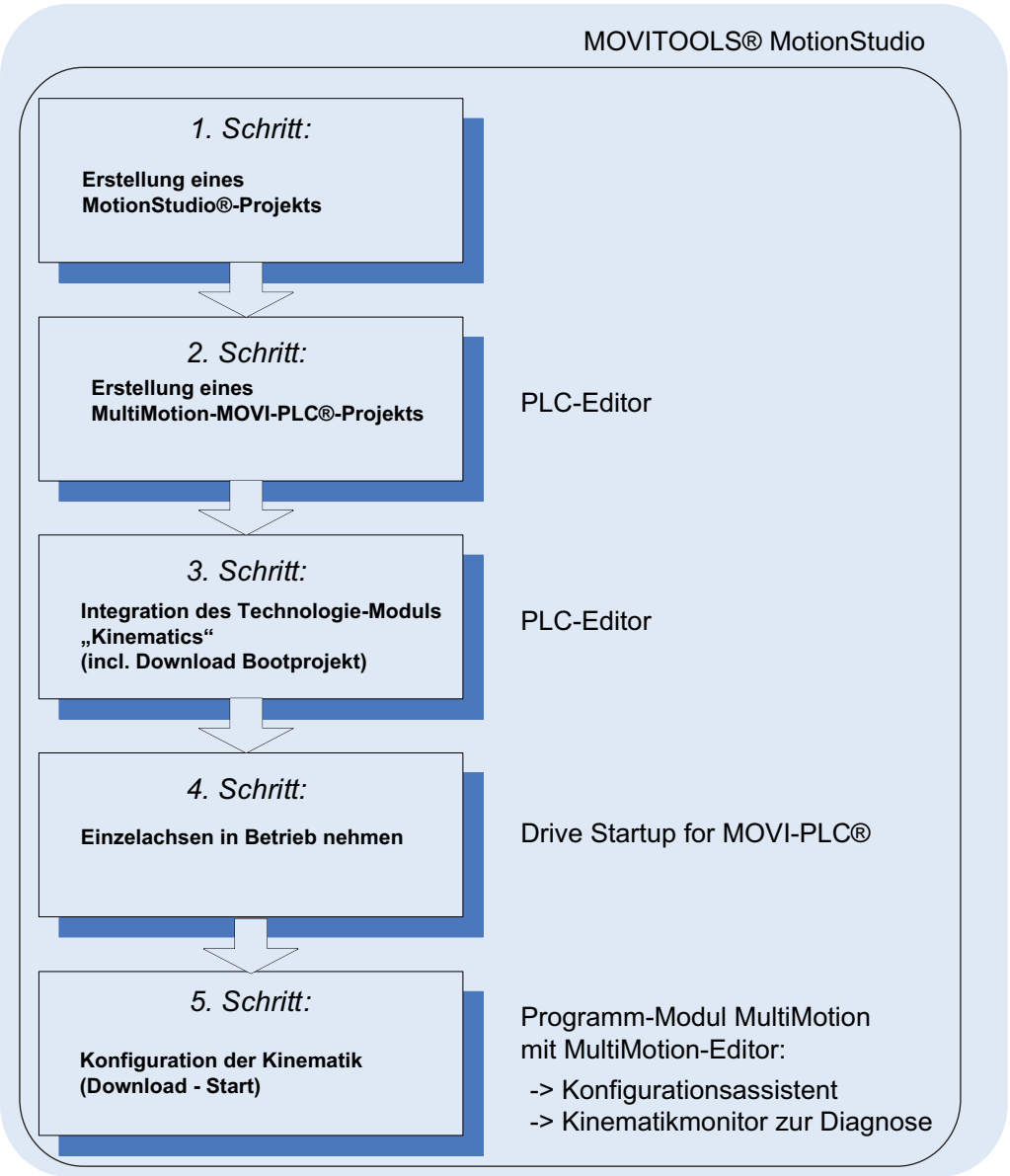
6.1 Ablauf

Für die Inbetriebnahme und Konfiguration benötigen Sie die Engineering-Software MOVITOOLS® MotionStudio.

Im Lieferumfang sind die folgenden Werkzeuge enthalten:

- PLC-Editor
- Drive Startup for MOVI-PLC®
- MultiMotion / MultiMotion Light mit folgenden Werkzeugen:
  - MultiMotion-Editor
  - Konfigurationsassistent

Sie benötigen alle 3 Werkzeuge für die Inbetriebnahme und Konfiguration. Die nachfolgende Darstellung zeigt den gesamten Ablauf:



9380451083

Zu Schritt 4:

- Markieren Sie den **Antrieb**, den sie in Betrieb nehmen möchten in der Netzwerksicht von MOVITOOLS® MotionStudio, und nehmen Sie ihn mit dem „Drive Startup“ aus dem Kontextmenü in Betrieb.
- Soll zunächst ausschließlich offline programmiert werden, also mit simulierten Antrieben, ist dieser Schritt nicht erforderlich.

Zu Schritt 5:

- Markieren Sie den **Controller** in der Netzwerksicht von MOVITOOLS® MotionStudio und starten Sie den Technologieeditor „MultiMotion“ / „MultiMotion Light“.

Zur Diagnose der konfigurierten Kinematik verwenden Sie die folgenden Werkzeuge:

- Kinematikmonitor im Technologieeditor „MultiMotion“ / „MultiMotion Light“
  - Hand-/Steuerbetrieb aktivierbar
- 3D-Simulation(startet automatisch mit dem Kinematikmonitor)
  - Zur Überprüfung der Bewegungsabläufe.
  - Abgleich der Konfiguration mit dem realen Roboter, denn erst bei Übereinstimmung ist die Konfiguration korrekt
- MessageHandler(aufrufbar aus dem Kontextemenü des Controllers)
  - Liefert ihnen detaillierte Klartextinformationen zu Fehlern und Warnungen.

Im Folgenden finden Sie eine detaillierte Beschreibung aller Schritte.

Nach dem erfolgreichen Abschluss der Konfiguration erstellen Sie Ihr Bewegungs- und Ablaufprogramm (siehe Kapitel "MOVI-PLC®-Programm").

6.2 Voraussetzungen

- Überprüfen Sie die Installation der Umrichter, den Anschluss der Geber und die Installation der MOVI-PLC® anhand der Installationshinweise in den Betriebsanleitungen und in den Feldbushandbüchern.
- Stellen Sie sicher, dass Sie für die zu konfigurierende Kinematik über eine ausreichende Anzahl an Technologiepunkten verfügen. Details hierzu entnehmen Sie dem Kapitel "Erforderliche Technologiepunkte".
- Spezielle, über die grundlegende Funktionalität hinausgehende Kinematikeinstellungen erfordern die Umstellung der MotionStudio®-Berechtigungsstufe auf "Advanced" oder die "erweiterte Konfiguration".

  Wie Sie die MotionStudio-Berechtigungsstufe ändern finden Sie im Kapitel "Wechseln der Motion-Studio-Berechtigungsstufe" (→ 🗎 39).

6.3 Schritt 1: Erstellung eines MotionStudio-Projekts

Weitere Informationen zu MOVITOOLS® MotionStudio finden Sie in der Dokumentation zu dieser Software.

Gehen Sie folgendermaßen vor:

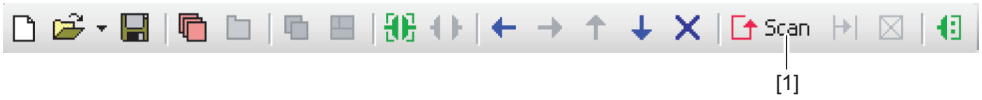
1. Verbinden Sie den Engineering-PC/-Laptop über Ethernet mit der MOVI-PLC®.

20211732 / DE – 07/2014

Achten Sie darauf, dass die IP-Adresse der Engineering-Schnittstelle und der angeschlossenen MOVI-PLC® im selben Subnetz liegen. Die IP-Adresse der MOVI-PLC® ist standardmäßig 192.168.10.4. Demnach sollte die IP-Adresse der Engineering-Schnittstelle auf 192.168.10.xxx eingestellt sein.

2. Versorgen Sie die MOVI-PLC® mit Spannung.
3. Starten Sie MOVITOOLS® MotionStudio.
4. Legen Sie ein neues Projekt im MOVITOOLS® MotionStudio an (Menü [Projekt] > [Neu]).

   Die Projektdatei wird im angegebenen Zielverzeichnis abgelegt.
5. Konfigurieren Sie die Engineering-Schnittstelle, mit der Sie auf die MOVI-PLC® zugreifen wollen (Menü [Netzwerk] > [Kommunikationsanschlüsse]).
6. Scannen Sie das Netzwerk (Verbindungsmodus "Online"), um die MOVI-PLC® zu finden (Symbol [Scan]).



   Die Steuerung an der Schnittstelle wird erkannt und automatisch in der Netzwerk-Sicht von MOVITOOLS® MotionStudios dargestellt.
7. Projektieren Sie die MOVI-PLC®, z. B., indem Sie das Gerät markieren und mit gedrückter Maustaste in die Projektsicht von MOVITOOLS® MotionStudio ziehen.

   Dabei erscheint ein Fenster, in dem Sie einen Gerätenamen eingeben müssen, anschließend werden Geräteparameter hochgeladen.

   Das projektierte Gerät wird in der Projektsicht dargestellt; ein blauer Kreis zeigt an, dass das Gerät projektiert ist. Beim Projektieren wird im Projektordner des MotionStudio im Unterverzeichnis "Devices" ein Unterordner mit dem Gerätenamen angelegt.

   Es empfiehlt sich, die Firmware dabei nicht mit einzubeziehen, da das Sichern sonst verhältnismäßig lange dauert. Eine entsprechende Einstellung können Sie im Menü [Einstellungen] > [Optionen] > [Sonstiges] vornehmen.
8. Erstellen Sie ein MOVI-PLC®-Projekt, wie im Folgenden beschrieben.

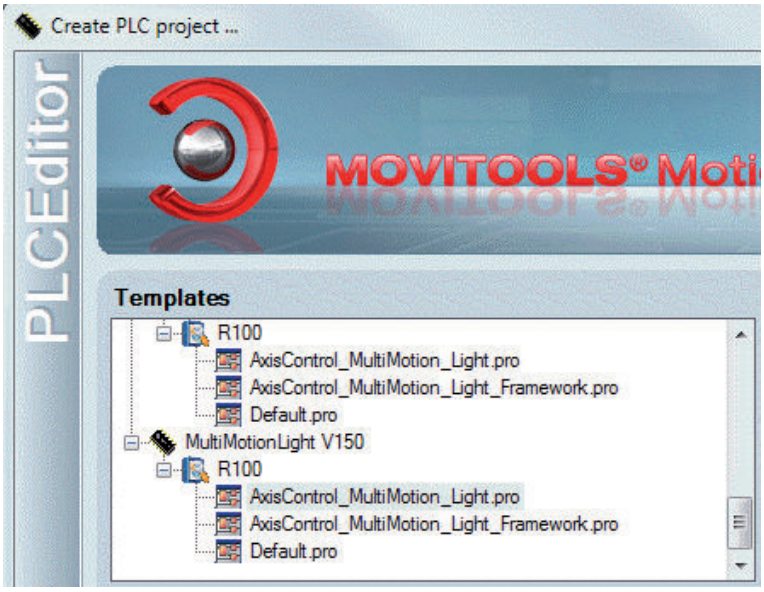
6.4 Schritt 2: Erstellung eines MOVI-PLC®-Projekts

Legen Sie ein MOVI-PLC®-Projekt mithilfe des PLC-Editors folgendermaßen an:

1. Starten Sie den Projektassistenten. Klicken Sie dazu mit der rechten Maustaste auf den Knotenpunkt der MOVI-PLC® (Kontextmenü [Programmierung] > [Neues PLC-Editor-Projekt erstellen]).
2. Wählen Sie im Projektassistenten die folgende Vorlage aus:

   `AxisControl_MultiMotion_Light`

20211732 / DE – 07/2014



9308425483

Hinweis zu den anderen auswählbaren Projekten:

- MultiMotion Full bringt für die Kinematiksteuerung keine zusätzlichen Vorteile.
- Wenn Sie ein Beispiel Applikations-Framework verwenden möchten, erstellen Sie ein Projekt mit dem Suffix _Framework.

3. Vergeben Sie einen Projektnamen und belassen Sie das standardmäßige Verzeichnis, da der MultiMotion-Editor die Symboldateien dort sucht.

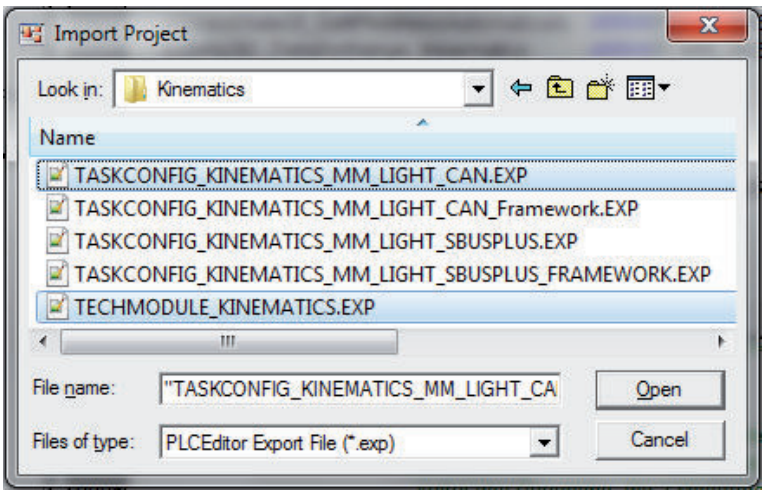
   Der PLC-Editor öffnet sich. Die Projektvorlage bildet die Basis des Steuerungsprogramms, in das der Anwender eigene Programm-Module ergänzen kann.

4. Integrieren Sie das Technologiemodul "Kinematics", wie im Folgenden beschrieben.

6.5 Schritt 3: Integration des Technologiemoduls

Gehen Sie folgendermaßen vor, um das Technologiemodul "Kinematics" in das MultiMotion- / MultiMotion-Light-MOVI-PLC®-Projekt zu importieren:

1. Wählen Sie den Menübefehl [Projekt] > [Importieren]:



9308509707

20211732 / DE – 07/2014

2. Importieren Sie aus dem Verzeichnis ...\"Ihr PLC-Projektordner"\TechModules\Kinematics die folgenden beiden Exportdateien:
   - TASKCONFIG_KINEMATICS_MM_xxx_xxx.EXP

     xxx ist abhängig vom ausgewählten Bussystem oder Gerät (CAN, SBUSPLUS, FDCadv) und abhängig vom erstellten MOVI-PLC®-Projekt (MultiMotion oder MultiMotion Light - mit oder ohne Beispiel Applikations-Framework)
   - TECHMODULE_KINEMATICS.EXP

HINWEIS

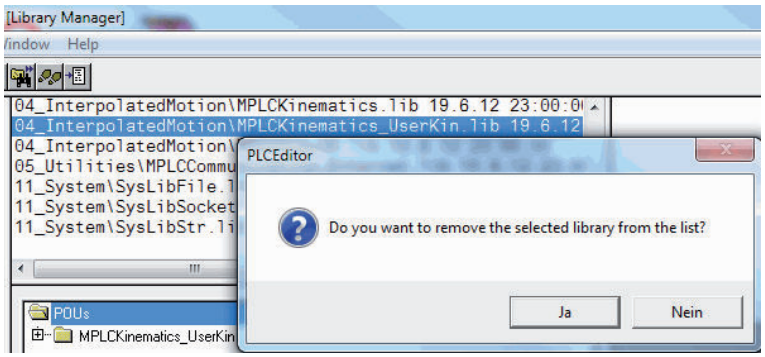
Beachten Sie bitte unbedingt, dass Sie für den Import in das Verzeichnis des PLC-Projekts navigieren, welches Sie im „Schritt 2: Erstellung eines MOVI-PLC®-Projekts (→ 🗎 27) erstellt haben. Das öffnende Fenster navigiert nicht automatisch zu diesem Verzeichnis. Wenn Sie die falschen Dateien importieren, ist das PLC-Projekt entweder nicht fehlerfrei übersetzbar oder funktioniert nicht korrekt.

3. Bestätigen Sie während des Imports das Ersetzen der Bausteine **Task_Main** und **Task_Priority**.

   Wenn Sie ein USER-Kinematikmodell verwenden (z. B. Articulated), müssen Sie im Bibliotheksverwalter die folgende Bibliothek löschen:

   "04_InterpolatedMotion\MPLCKinematics_UserKin.lib"

   und die entsprechende UserKin-Bibliothek von SEW-EURODRIVE hinzufügen.



9308559115

4. Übersetzen Sie das Projekt und laden Sie es in die MOVI-PLC® (Menü [Online] > [Einloggen] und Menü [Projekt] > [Übersetzen]).

   Die Projektvorlage kann im ersten Schritt ohne Veränderungen gestartet werden.
5. Erzeugen Sie ein Boot-Projekt (Menü [Online] > [Bootprojekt erzeugen]).

HINWEIS

Beachten Sie, dass die Task-Konfiguration beim Import ersetzt wird.

Wenn Sie kein neues MOVI-PLC®-Projekt erstellen, sondern das Technologiemodul "Kinematics" in ein bestehendes MOVI-PLC®-Projekt mit bereits geänderter Task-Konfiguration integrieren möchten, sollten Sie die Task-Konfiguration manuell erweitern. Hierzu können Sie z. B. das Technologiemodul "Kinematics" in ein neues Projekt importieren und die entsprechenden Task-Einträge aus diesem Projekt kopieren oder gemäß der Beschreibung in Kapitel "Task-Konfiguration" manuell hinzufügen.

6.6 Schritt 4: Einzelachsen in Betrieb nehmen

Gehen Sie folgendermaßen vor, um die Einzelachsen in Betrieb zu nehmen:

1. Starten Sie die Engineering-Software "MOVITOOLS® MotionStudio".
2. Führen Sie für alle Antriebe den Inbetriebnahmeassistenten "DriveStartup for MOVI-PLC" aus.

6.7 Schritt 5: Konfiguration der Kinematik

6.7.1 Neues Technologiemodul einfügen

Gehen Sie folgendermaßen vor, um eine neues Technologie-Modul "Kinematics" einzufügen:

1. Klicken Sie auf der Startseite des MultiMotion-Editors auf die Schaltfläche [Neue Konfiguration erstellen] [1].

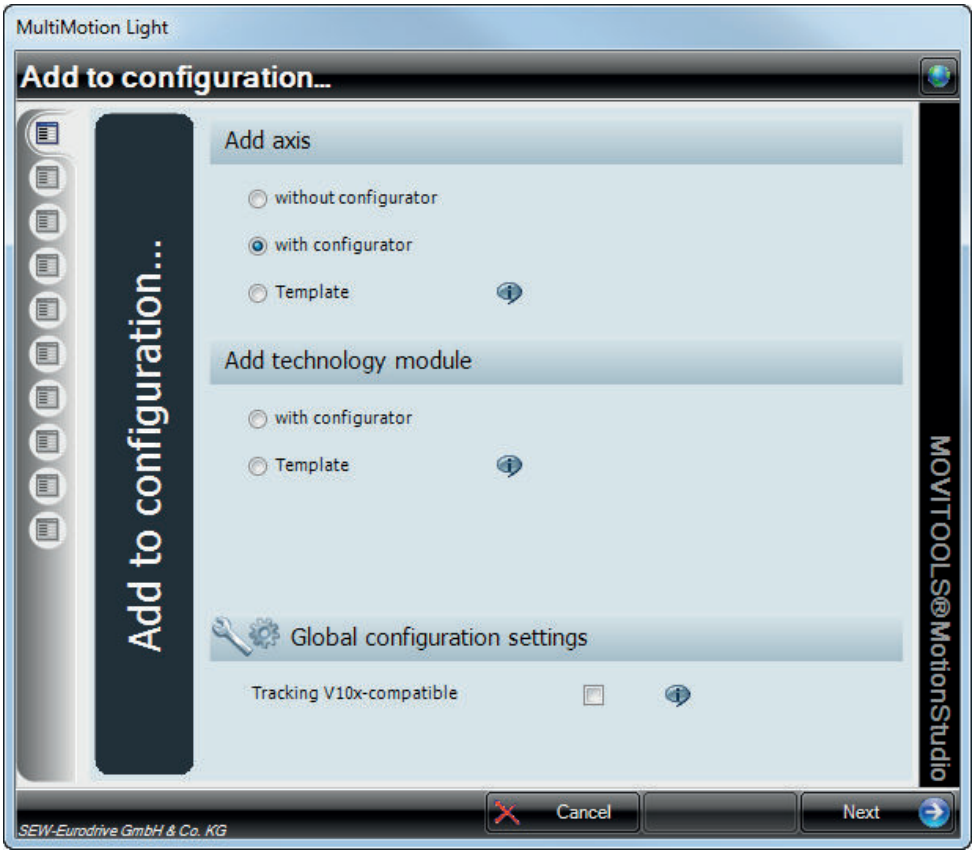
9409047563

Ein Fenster zur Auswahl des Technologiemoduls öffnet sich:

20211732 / DE – 07/2014

9463746955

2. Wählen Sie als Technologiemodul "Kinematics" mit Konfigurator und fahren Sie fort mit [Weiter].



9463750667

Es öffnet sich der Konfigurationsassistent.

3. Folgen Sie den Anweisungen des Assistenten. Details entnehmen Sie den folgenden Abschnitten.

   Wenn Sie Ihre Eingaben erfolgreich abgeschlossen haben. schließt der Assistent und fügt das konfigurierte Technologiemodul „Kinematics“ ein. Sie können es anschließend jederzeit mit dem MultiMotion-Editor aufrufen und bearbeiten.

6.7.2 Auswahl von "Kinematics" oder "HandlingsKinematics"

Die Technologiemodule Kinematics und HandlingKinematics werden im Kapitel „Auswahl von Kinematics oder HandlingKinematics (→ 🗎 21)“ gegenübergestellt.

Die Besonderheiten des Technologiemoduls „HandlingKinematics“ sind im Handbuch „Technologiemodul HandlingKinematics für MultiMotion / MultiMotion Light“ dokumentiert.

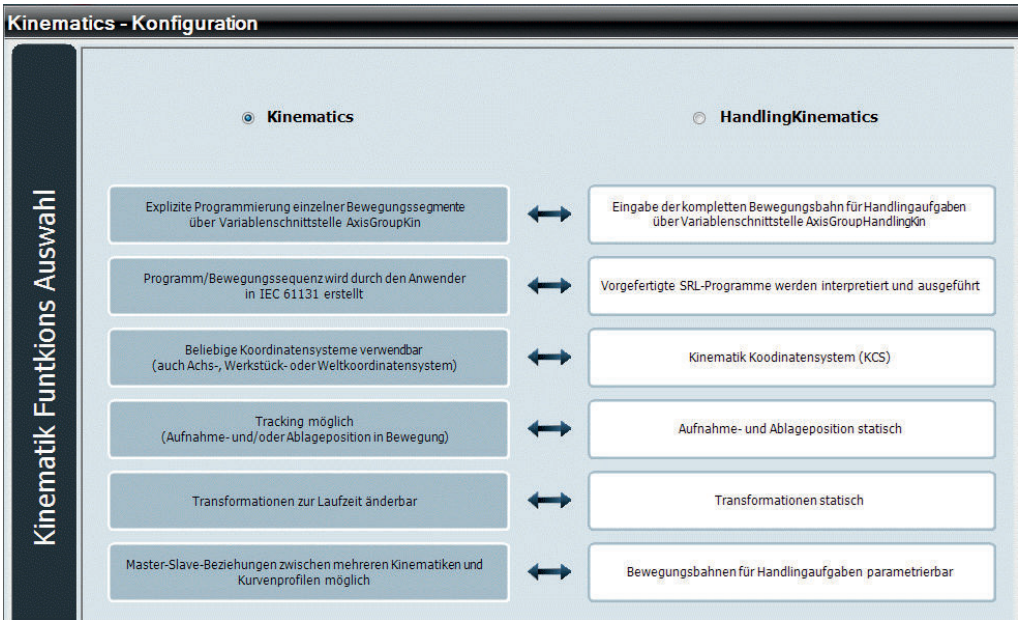
Da das Technologiemodul HandlingKinematics auf dem Technologiemodul Kinematics aufsetzt, sind grundlegende kinematische Beschreibungen und Eigenschaften für beide Technologiemodule gültig.

Das vorliegende Handbuch beschreibt zusätzlich die umfassenden Funktionalitäten des Technologiemoduls Kinematics.

Gehen Sie folgendermaßen vor, um das Technologiemodul auszuwählen:

- Wählen Sie zur Konfiguration des Technologiemoduls Kinematics die Schaltfläche „Kinematics“ aus:

20211732 / DE – 07/2014

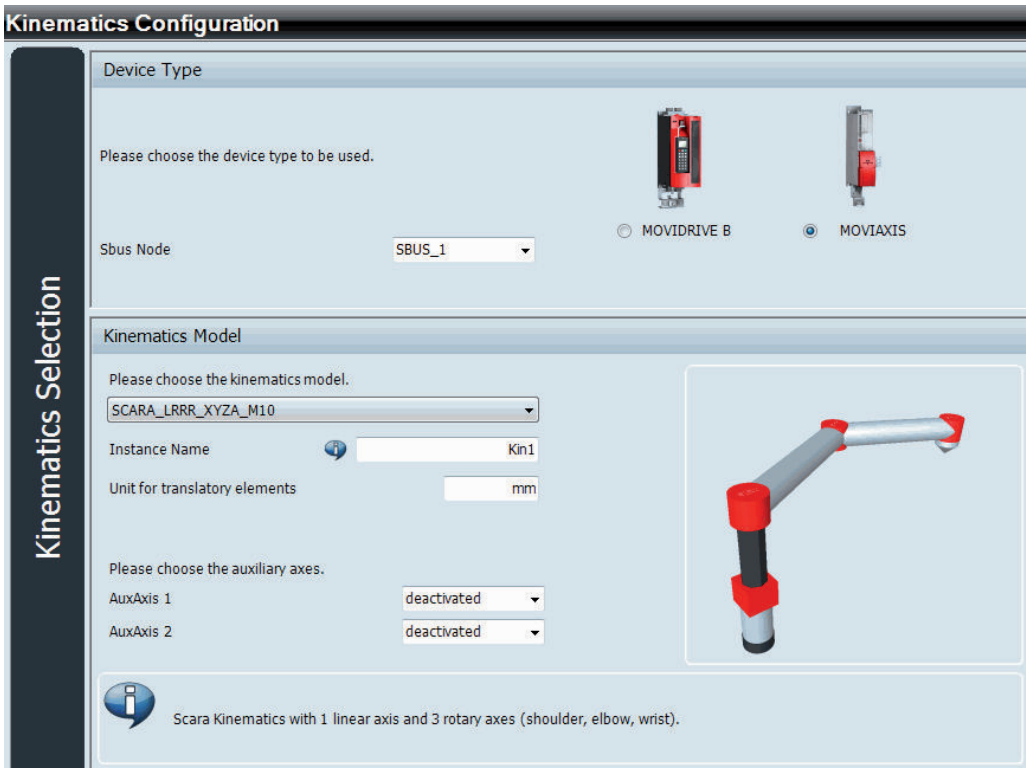

Kinematics - Konfiguration
Kinematik Funtkions Auswahl
Kinematics
HandlingKinematics
Explizite Programmierung einzelner Bewegungssegmente über Variablenschnittstelle AxisGroupKin
Eingabe der kompletten Bewegungsbahn für Handlingaufgaben über Variablenschnittstelle AxisGroupHandlingKin
Programm/Bewegungssequenz wird durch den Anwender in IEC 61131 erstellt
Vorgefertigte SRL-Programme werden interpretiert und ausgeführt
Beliebige Koordinatensysteme verwendbar (auch Achs-, Werkstück- oder Weltkoordinatensystem)
Kinematik Koodinatensystem (KCS)
Tracking möglich (Aufnahme- und/oder Ablageposition in Bewegung)
Aufnahme- und Ablageposition statisch
Transformationen zur Laufzeit änderbar
Transformationen statisch
Master-Slave-Beziehungen zwischen mehreren Kinematiken und Kurvenprofilen möglich
Bewegungsbahnen für Handlingaufgaben parametrierbar

12497284747

6.7.3 Auswahl Gerätetyp und Kinematikmodell

Gehen Sie folgendermaßen vor, um Gerätetyp und Kinematikmodell auszuwählen:

1. Wählen Sie den Gerätetyp des Umrichters aus, sowie den Bustyp.



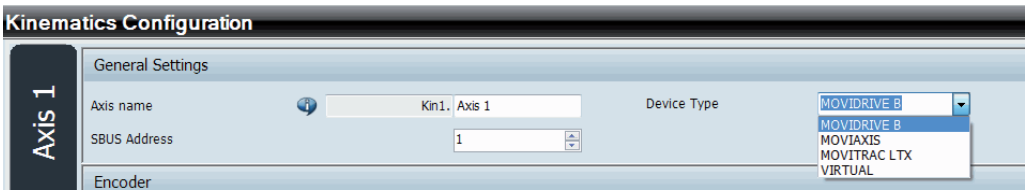
9308358283

2. Wählen Sie das gewünschte Kinematikmodell aus. Details hierzu finden Sie im Kapitel "Kinematikmodelle" (→ 🗎 57).

Achten Sie, bei der Auswahl des Kinematikmodells darauf, dass die Speicherkarte über eine ausreichende Anzahl an Technologiepunkten verfügt. Details hierzu finden Sie im Kapitel "Erforderliche Technologiestufe" (→ 🗎 13).

Einrichten von individuellen Gerätetypen

Wenn ein Gerätetyp von den anderen in der Kinematikinstanz eingesetzten Umrichtern abweicht, können Sie mit MotionStudio-Berechtigungsstufe "Advanced" in den "Achseinstellungen" individuelle Typen einstellen.

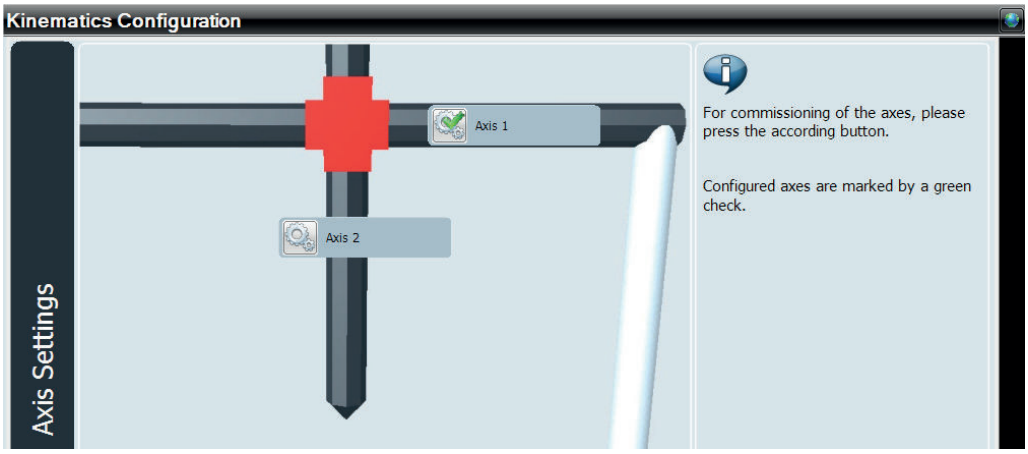


9308351115

6.7.4 Konfiguration der Achsen

Um die Achsen zu konfigurieren, fahren Sie im Konfigurationsassistenten fort:

1. Klicken Sie auf die Schaltfläche der zu konfigurierenden Achse.


Kinematics Configuration
Axis Settings
Axis 1
Axis 2
For commissioning of the axes, please press the according button.
Configured axes are marked by a green check.

9400622475

2. Nehmen Sie nacheinander die folgenden Achseinstellungen vor:
   - Allgemeine Einstellungen (Achsnamen und SBUS-Adresse)
   - Geber (Quelle Istposition)
   - Skalierung (Übersetzungsverhältnis etc.)
   - Referenzfahrtparameter
   - Systemgrenzen (Schnellstopp-Rampe, Maximale Motordrehzahl, Schleppfehlerfenster, Software-Endschalter etc.)
   - Rampen für Achsen-Tippbetrieb (**KIN_JOG_AXIS**)
   - Verwendung der digitalen Eingänge und Hardware-Endschalter

   Zur Bestätigung, dass die Achse vollständig eingestellt ist, erscheint ein Haken an der Schaltfläche.
3. Wiederholen Sie den Vorgang für alle weiteren Achsen.

Die Anpassung des Kinematikmodells an die real vorliegenden Achseinstellungen finden Sie in den folgenden Abschnitten.

20211732 / DE – 07/2014

Wichtiger Hinweis zur Einstellung der Software-Endschalter

HINWEIS

Die Software-Endschalter werden in Bezug auf den Nullpunkt und die Bewegungsrichtung der jeweiligen **realen** Achse eingestellt.

Der Nullpunkt ist die Motorstellung bei 0 Inkrementen und die Bewegungsrichtung ist positiv bei zunehmenden Motorinkrementen.

HINWEIS

Die Software-Endschalter der Achsen werden in jeder Betriebsart geprüft.

Sie können auch ignoriert werden durch Setzen der Variablen *AxisGroup-Kin.Inst[..].In.General.Ignore.SWLS.Axis*, so dass eine Bewegung außerhalb der Software-Endschalter der Achsen möglich ist.

6.7.5 Anpassung der Modellachsen an reale Achsen

Damit die Kinematiksteuerung korrekt funktioniert, müssen Sie das ausgewählte Kinematikmodell an die realen Achsen anpassen.

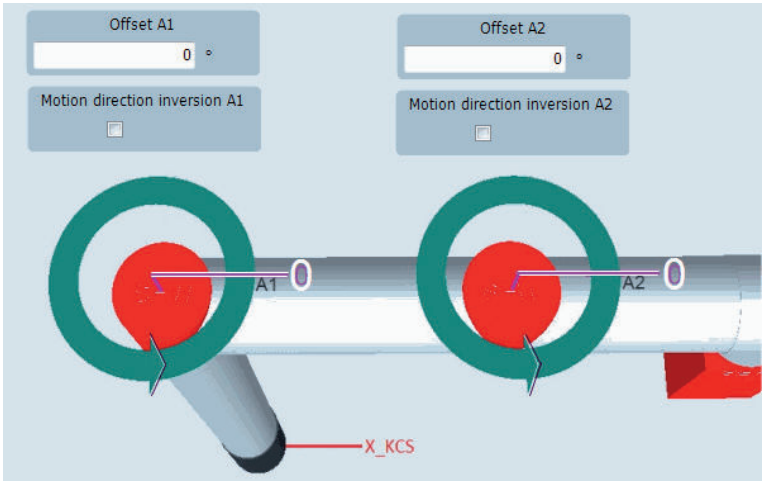
Beispielsweise werden bei Delta- oder SCARA-Kinematiken trigonometrische Berechnungen ausgeführt, die nur korrekt funktionieren, wenn die hierbei verwendeten Winkel zwischen den zueinander verdrehten Armteilen den korrekten Bezugspunkt und die korrekte Richtung haben.

Entsprechend müssen Sie hier für eine Achse eine Drehrichtungsumkehr einstellen, wenn die positive Drehrichtung der Modellachse von der positiven Bewegungsrichtung der realen Achse (Zunahme der Motorinkremente) abweicht.

Zusätzlich müssen Sie für eine Achse einen Offset eintragen, wenn die Nullstellung des Modells von dem Achsennullpunkt abweicht. Dieser Offset wird in den zuvor konfigurierten Anwendereinheiten angegeben. Er gibt an wo die Nullstellung des Modells liegt, in Bezug auf den realen Achsennullpunkt entlang der realen Bewegungsrichtung.

Anzeige der Nullstellung des Modells und positive Bewegungsrichtung

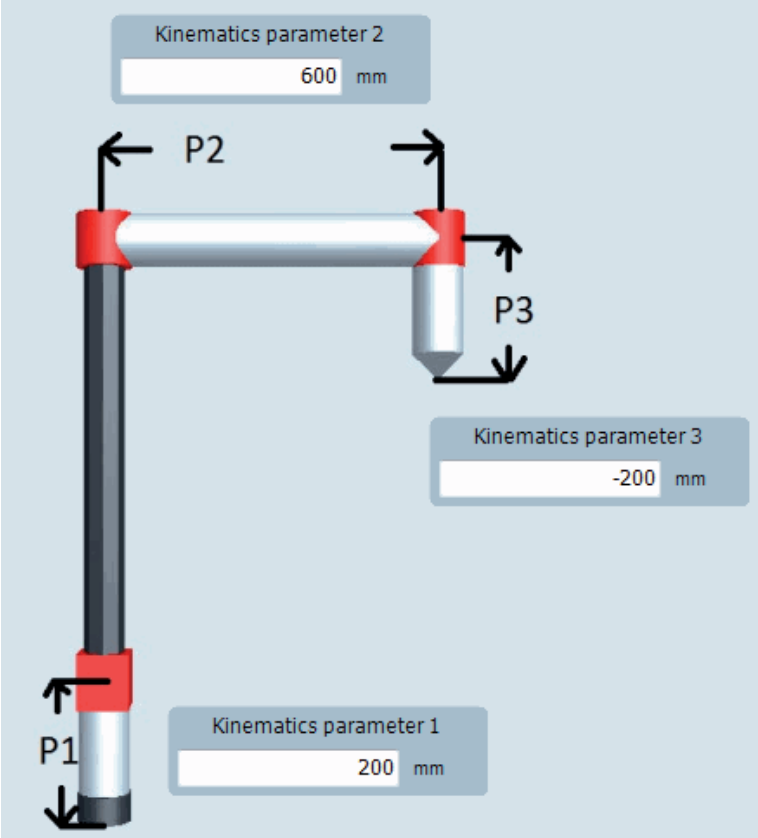
Die Nullstellung, sowie die positive Bewegungsrichtung des Modells werden mithilfe von Pfeilen in entsprechenden Abbildungen im Konfigurationsassistenten angezeigt:



9308407691

6.7.6 Einstellung der Kinematikparameter

Beispiel: Einstellung der Abstände zwischen den Gelenken der kinematischen Kette



9308378635

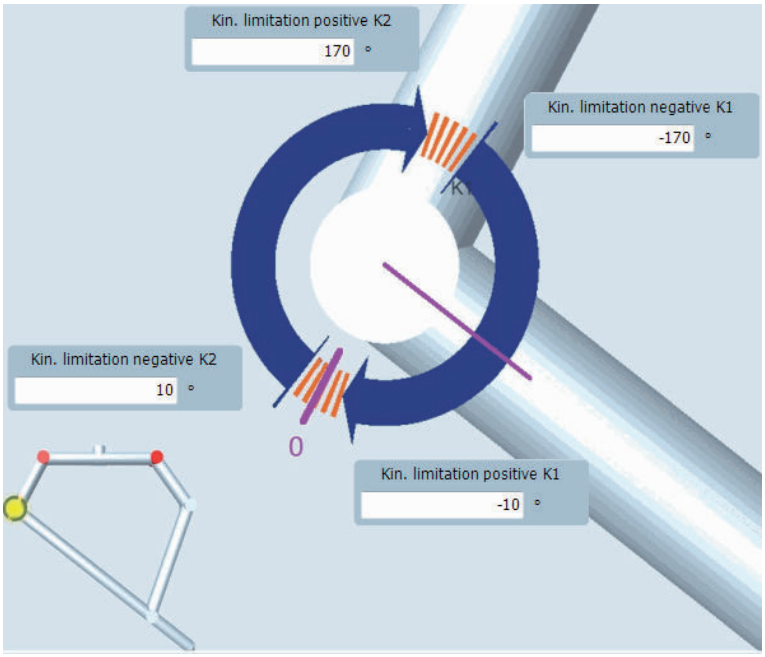
6.7.7 Einstellung der Kinematischen Software-Endschalter

Bei einigen Kinematikmodellen gibt es neben den Software-Endschaltern der Achsen und den kartesischen Software-Endschaltern auch eine dritte Art von Software-Endschaltern: Die kinematischen Software-Endschalter

Einstellung gültiger / verbotener Winkelbereiche

20211732 / DE – 07/2014

Mithilfe der kinematischen Software-Endschalter können Sie beispielsweise gültige bzw. verbotene Winkelbereiche von Gelenken definieren, denen nicht direkt ein Antrieb zugeordnet ist:



9308386571

HINWEIS

Die kinematischen Software-Endschalter werden in jeder Betriebsart geprüft.

Sie können auch ignoriert werden durch Setzen der Variablen *AxisGroup-Kin.Inst[..].In.General.Ignore.SWLS.Kin*, so dass eine Bewegung außerhalb der kinematischen Software-Endschalter möglich ist.

6.7.8 Kartesische Einstellungen

Hier nehmen Sie für die kartesischen Freiheitsgrade (X, Y, Z, A, B, C) u. a. die folgenden Einstellungen vor:

- Kartesische Software-Endschalter
- Rampen für kartesischen Tippbetrieb (**KIN_JOG_CART**)
- Schnellstopp-Rampen

HINWEIS

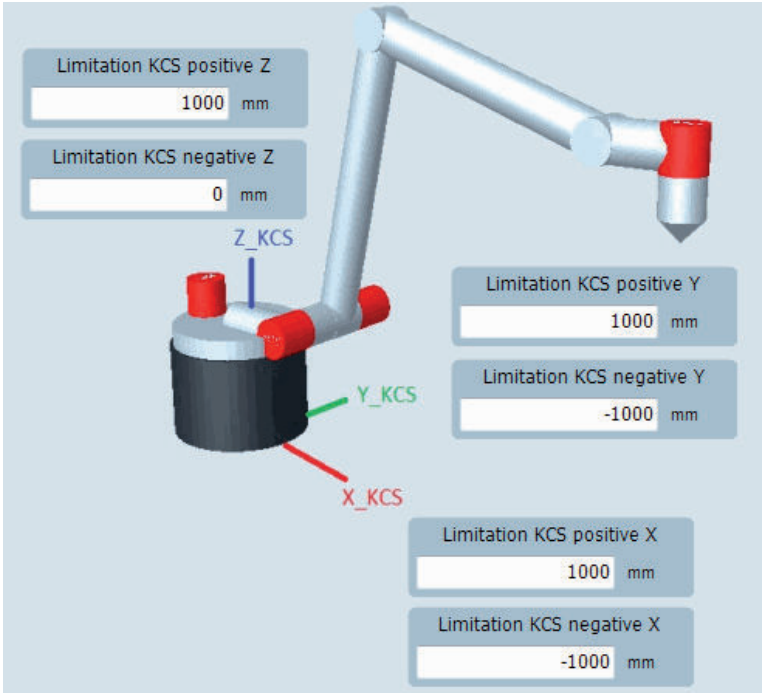
Die kartesischen Software-Endschalter werden in jeder Betriebsart geprüft.

Sie können auch ignoriert werden durch Setzen der Variablen *AxisGroup-Kin.Inst[..].In.General.Ignore.SWLS.Cart*, so dass eine Bewegung außerhalb der kartesischen Software-Endschalter möglich ist.

Die kartesischen Schnellstopp-Rampen werden zum kartesischen Abbremsen der Bewegung im aktuell eingestellten Koordinatensystem verwendet, wenn sich die Kinematiksteuerung bei Auftreten eines Kinematikfehlers oder bei Setzen von *AxisGroup-Kin.Inst[..].In.General.Kin.RapidStop* in der Betriebsart **KIN_JOG_CART** oder **KIN_TARGET_CART** befindet.

20211732 / DE – 07/2014

Beispiel: Einstellung der kartesischen Software-Endschalter



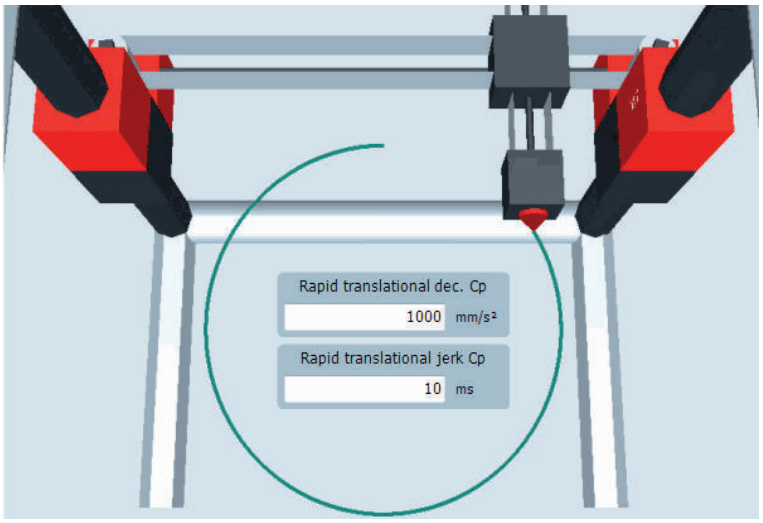
9308331659

6.7.9 ContinuousPath-Einstellungen

Hier stellen Sie die Schnellstopp-Rampen für die ContinuousPath-Translation ein.

Die CP-Schnellstopp-Rampen werden zum Abbremsen der Bewegung entlang der CP-Bahn im aktuell eingestellten Koordinatensystem verwendet, wenn sich die Kinematiksteuerung bei Auftreten eines Kinematikfehlers oder bei Setzen von *AxisGroupKin.Inst[..].In.General.Kin.RapidStop* in einer ContinuousPath-Betriebsart befindet.

Beispiel: Einstellung der Schnellstopp-Rampen



9308337931

20211732 / DE – 07/2014

6.7.10 Weitere Einstellungen

Hier stellen Sie die IP-Adresse des Engineering-PC´s ein, auf dem die 3D-Simulation läuft.

6.7.11 Standardbelegung von Eingangsvariablen

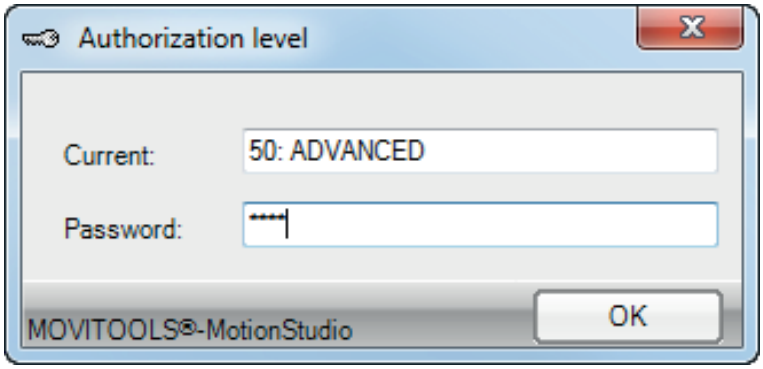
Die Defaultwerte ausgewählter Eingangsvariablen werden beim Starten der MOVI-PLC® zugewiesen. Zum Ändern der Werte verwenden Sie entweder die „Erweiterte Konfiguration“ oder wechseln Sie die MotionStudio-Berechtigungsstufe "Advanced".

Wechseln der Motion-Studio-Berechtigungsstufe

Gehen Sie folgendermaßen vor, um die MotionStudio-Berechtigungsstufe zu wechseln:

1. Wählen Sie in MOVITOOLS® MotionStudio den Menübefehl [Einstellungen] > [Berechtigungsstufen].

   Es öffnet sich ein Fenster zur Eingabe des Passworts für die gewünschte Berechtigungsstufe:



9455251211

2. Geben Sie in die untere Zeile "5249" ein.

   In der darüberliegenden Zeile wechselt die Berechtigungsstufe zu "Advanced".

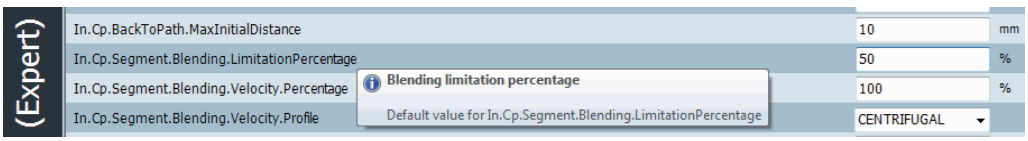
3. Starten Sie den Technologieeditor MultiMotion / MultiMotion Light neu.

Ändern der Standardbelegung

Beim Durchlaufen des Assistenten zur Konfiguration der Kinematik erhalten Sie ein zusätzliches Fenster in dem Sie in Listenform detaillierten Zugriff auf alle Konfigurationseinstellungen haben.

Sie sollten im oberen Bereich der Liste ohne Rücksprache mit SEW-EURODRIVE® keine **Änderungen** vornehmen.

Am Ende der Liste sind die Parameter für die Standardbelegung ausgewählter Eingangsvariablen aufgeführt. Diese Standardbelegung können Sie bei Bedarf an Ihre Erfordernisse anpassen.



9308344331

Die Zuweisungen erfolgen beim Start der MOVI-PLC®, sobald die Variable *AxisGroupKin.ConfigDataAvailable* = TRUE ist.

(Beachten Sie hierzu den Hinweis im folgenden Abschnitt "Download").

Die folgende Tabelle zeigt die Eingangsvariablen mit Ihren Standardwerten:

Eingangsvariable	Standardwert	Einheit
In.Transform.Tool[1..6]	0	[Anwendereinheit]
In.Transform.WCS_KCS[1..6]	0	[Anwendereinheit]
In.Cp.BackToPath.MaxInitialDistance	10	[Anwendereinheit]
In.Cp.Segment.Blending.LimitationPercentage	50	%
In.Cp.Segment.Blending.Velocity.Percentage	100	%
In.Cp.Segment.Blending.Velocity.Profile	HANDLING	-
In.Cp.Path.FidelityPercentage	95	%
In.Cp.Segment.Circ.EllipticDistortionFactor	1	-
In.Cp.Path.VelocityPercentage	100	%
In.Target.RapidDynamicsAtOvershooting	FALSE	-
In.Target.CoordSys	KCS	-
In.General.Kin.OverridePercentage	10	%
In.General.Ignore.Inverter[1..8]	FALSE	-
In.Jog.CoordSys	KCS	-
In.Jog.VelocityPercentage	10	%
In.Simu3D.Enable	TRUE	-

Die Bedeutung der Eingangsvariablen entnehmen Sie dem Kapitel: "Variablenschnittstelle *AxisGroupKin.Inst[..]*".

6.7.12 Download

Beim Herunterladen der Konfiguration wird die MOVI-PLC® neu gestartet. Dabei übernimmt sie Konfigurationsdaten aus XML-Dateien, die sich auf der Speicherkarte befinden.

HINWEIS

Stellen Sie sicher, dass Ihr Anwenderprogramm NICHT startet, bevor *AxisGroupKin.ConfigDataAvailable* = TRUE gilt.

Ansonsten stehen die Konfigurationsdaten und die Defaultbelegungen für die Eingangsvariablen noch nicht zur Verfügung. Eventuelle Zuweisungen aus dem Anwenderprogramm (z. B. Override) werden noch überschrieben.

20211732 / DE – 07/2014

7 Diagnose

7.1 Kinematikmonitor

7.1.1 Bedienoberfläche

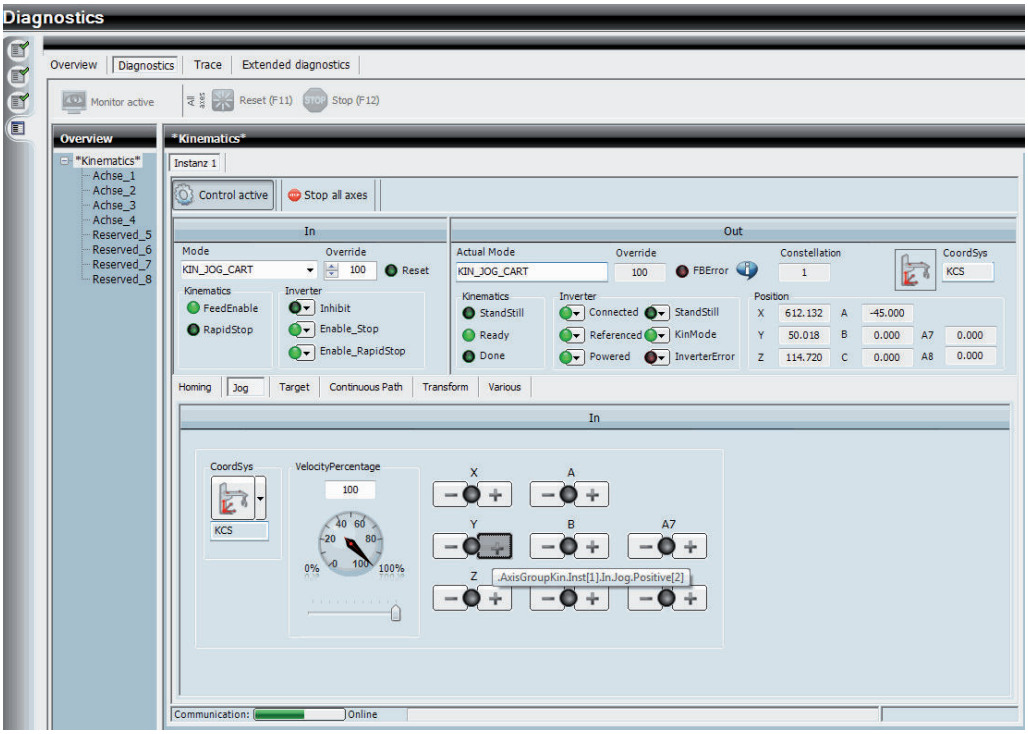
Kinematikmonitor aufrufen

Gehen Sie folgendermaßen vor, um den Kinematikmonitor aufzurufen:

1. Wechseln Sie zur Startseite des MultiMotion- / MultiMotion Light-Editors.
2. Klicken Sie auf die Schaltfläche [Diagnose]

   Die Konfiguration des Technologiemoduls wird von der Speicherkarte des Controllers auf die Diagnoseoberfläche übertragen. Dieser Vorgang kann einige Sekunden dauern.

   Der Kinematikmonitor öffnet sich:



9308751499

3. Sie finden die folgenden Diagnosebereiche als Registerkarten:

Diagnosebereich	Funktion	Bemerkung
Überblick	Detaildiagnose der verschiedenen Module	Die Informationen kommen direkt aus dem Controller und sind feldbusunabhängig.
Diagnose	Zeigt die Ein- und Ausgangssignale der Variablenschnittstelle *AxisGroupKin.Inst[n]* übersichtlich angeordnet an. Details hierzu entnehmen Sie dem Kapitel "Variablenschnittstelle AxisGroupKin.Inst[..]". (→ 🗎 88)	Sie können zwischen den Betriebsarten "Monitormodus" und "Steuermodus" wählen.

Diagnosebereich	Funktion	Bemerkung
Trace	Aufzeichnung verschiedener Prozesssignale (Geschwindigkeit, Position der Achse etc.)	Sie können bis zu 4 Kanäle gleichzeitig aufzeichnen.
Erweiterte Diagnose	Die erweiterte Diagnose dient als Expertendiagnose.	Nutzen Sie die erweiterte Diagnose nur nach Rücksprache mit SEW.

Registerkarten herauslösen

Sie können die Fenster von Registerkarten "herauslösen" und separat darstellen. Gehen Sie folgendermaßen vor:

1. Halten Sie die Steuerungstaste gedrückt und doppelklicken Sie auf den In-/Out-Bereich in der betreffenden Registerkarte.

   Das Fenster wird "herausgelöst" und separat dargestellt.

2. Zum Beenden, schließen Sie das Fenster.

Tooltip zu Variablen anzeigen

Die Namen aller Variablen, die Sie im Kinematikmonitor einstellen können, erhalten Sie als Tooltip. Mit dem Namen finden Sie detaillierte Informationen im Kapitel "Variablenschnittstelle AxisGroupKin.Inst[..]" (→ 🗎 88)

Zusätzlich stellt die Anzeige der entsprechenden Variablen eine Hilfestellung zur Programmierung dar.

20211732 / DE – 07/2014

7.1.2 Monitor- und Steuerbetrieb

Zur Diagnose verfügt das Technologiemodul "Kinematics" über einen eigenständigen Monitor- und Steuerbetrieb:

- **Monitorbetrieb:** Im Monitorbetrieb werden die Diagnosedaten des angewählten Technologiemoduls angezeigt. Dabei greift der Monitor lediglich lesend auf die entsprechenden Schnittstellen zu.

  Wenn Sie sich im Monitorbetrieb befinden, steht auf der Schaltfläche oben links [Monitor aktiv]. Wenn Sie auf diese Schaltfläche klicken, wechseln Sie in den Steuerbetrieb.

  ACHTUNG! Wenn Sie in den Steuerbetrieb wechseln, muss die Anlage in einem sicheren Zustand sein. Eine entsprechende Meldung muss bestätigt werden.

- **Steuerbetrieb:** Im Steuerbetrieb greift der Monitor schreibend und lesend auf die Schnittstellen des Technologiemoduls zu. Dabei ist es möglich, direkt an der Schnittstelle Steuersignale zu setzen und rückzusetzen sowie Variablenwerte einzustellen, z. B. Position, Geschwindigkeit.

  Wenn Sie sich im Steuerbetrieb befinden, steht auf der Schaltfläche oben links [Steuern aktiv].

  Wenn Sie auf diese Schaltfläche klicken, wechseln Sie wieder in den Monitorbetrieb. Bevor Sie den Monitorbereich verlassen können, z. B. um in den Konfigurationsbereich zu wechseln, muss der Steuerbetrieb wieder deaktiviert werden.

  Bei aktivem Steuerbetrieb wird beim Technologiemodul "Kinematics" die Variable *AxisGroupKin.HMI.HMIControl* in der globalen Variablenstruktur gesetzt (TRUE).

  In diesem Fall müssen Sie sicherstellen, dass Ihr Anwenderprogramm nicht ausgeführt wird bzw. die globale Variable AxisGroupKin nicht beschreibt.In den mit dem Technologiemodul importierten Beispielprogrammen ist die entsprechende Fallunterscheidung enthalten, siehe **PRG_User_Kinematics_Examples** im Kapitel "Beispielprogramme" (→ 🗎 83).

20211732 / DE – 07/2014

In den Steuermodus wechseln

Gehen Sie folgendermaßen vor, um in den Steuermodus zu wechseln:

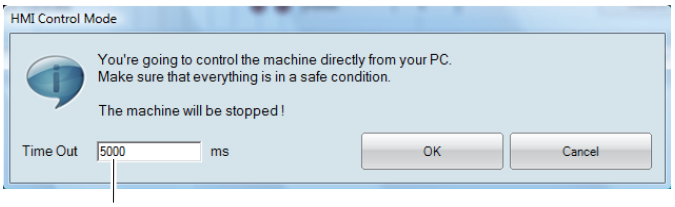
1. ⚠**GEFAHR!** Unvorhergesehene Bewegung der Maschine beim Wechsel vom Monitormodus in den Steuermodus und umgekehrt.

   Tod oder schwere Körperverletzung.

   - Stellen Sie sicher, dass ein selbsttätiges Anlaufen oder Stoppen der Maschine keine Gefahr für Personen und Geräte verursacht.
   - Stellen Sie sicher, dass die Maschine in einem sicheren Zustand ist.

2. Klicken Sie auf die Schaltfläche [Monitor aktiv], um in den Steuermodus zu wechseln.

   Das Programm fordert Sie auf eine Zeit für das Timeout [1] anzugeben:



9470720779

3. Tragen Sie einen geeigneten Wert für das Timeout ein und bestätigen Sie mit [OK].

   Die Schaltfläche zeigt jetzt den Steuermodus an:

9470737035

20211732 / DE – 07/2014

7.1.3 Einstellung für kinematische Betriebsart

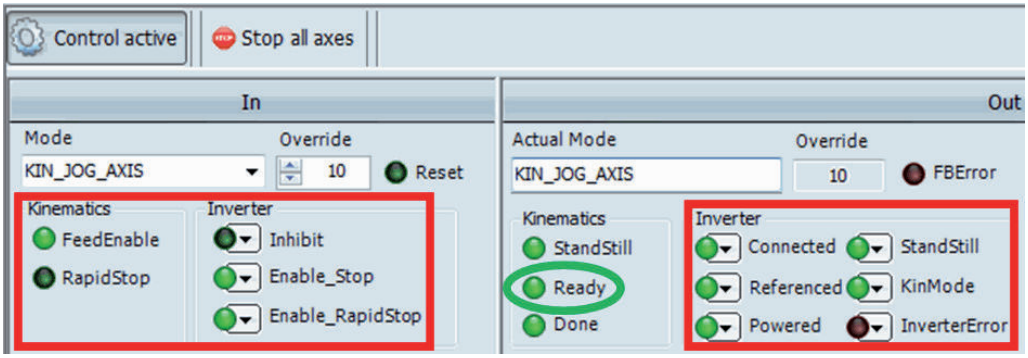
1. Damit eine kinematische Betriebsart, z. B. **KIN_JOG_AXIS** ausführbar ist, stellen Sie die folgenden Eingangssignale (Variablen) im In-Bereich ein:

In-Bereich	Einstellwert
Inhibit	FALSE
Enable_Stop	TRUE
Enable_RapidStop	TRUE
FeedEnable	TRUE
RapidStop	FALSE

2. Wenn alle Eingangsvariablen richtig gesetzt sind, vergewissern Sie sich, dass die Ausgangssignale folgende Werte anzeigen:

Out-Bereich	Anzeigewert	Bemerkung
Connected	TRUE	-
Referenced	TRUE	-
Powered	TRUE	-
InverterError	FALSE	-
Inverter.StandStill	TRUE	Beim Wechsel in eine kinematische Betriebsart
	FALSE	Bei Bewegung nach dem Wechsel in eine kinematische Betriebsart
Inverter.KinMode	TRUE	Nach dem Wechsel in eine kinematische Betriebsart

Die folgende Darstellung zeigt den In- und Out-Bereich, wenn alle Voraussetzungen für eine kinematische Betriebsart erfüllt sind:



9308883467

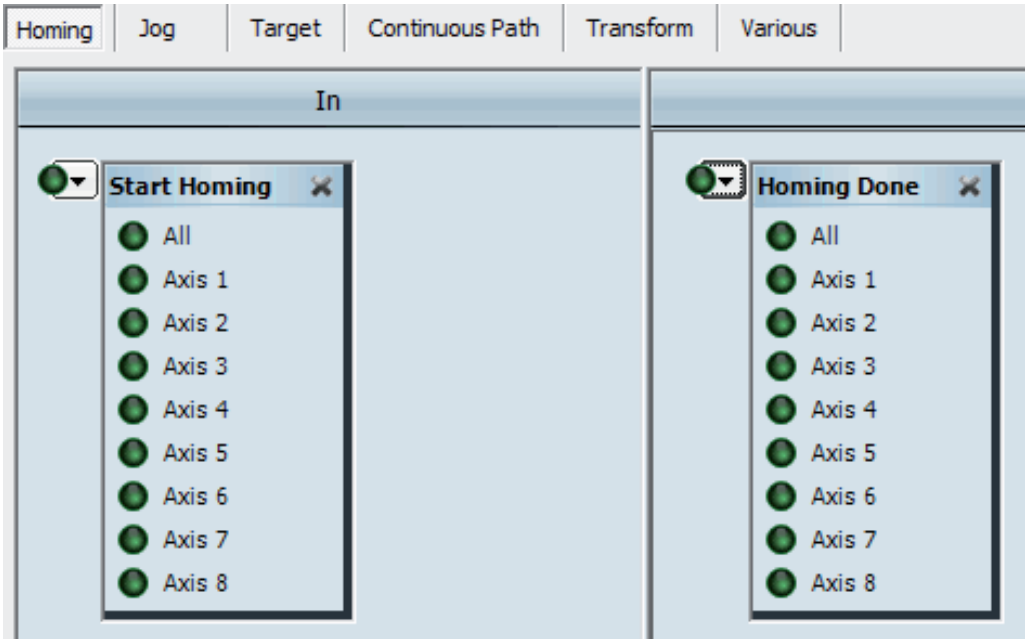
HINWEIS

Im Fehlerfall erhalten Sie detaillierte Klartexinformationen in der Registerkarte Message des Kinematik-Monitors oder in übersichtlicher Darstellung im Diagnosewerkzeug"MessageHandler (→ 🗎 55)“.

7.1.4 Referenzieren der Achsen

1. Stellen Sie die Betriebsart **KIN_AM_HOMING** ein.
2. Wechseln Sie zur Registerkarte "Homing" (In-Bereich)
3. Referenzieren Sie wahlweise alle oder bestimmte Achsen, indem Sie auf das jeweilige LED-Symbol klicken.

   Das Referenzieren erfolgt in der zuvor konfigurierten Betriebsart (z. B. Referenznocken links).



9308642315

Das Ergebnis der Aktion wird Ihnen im Out-Bereich angezeigt.

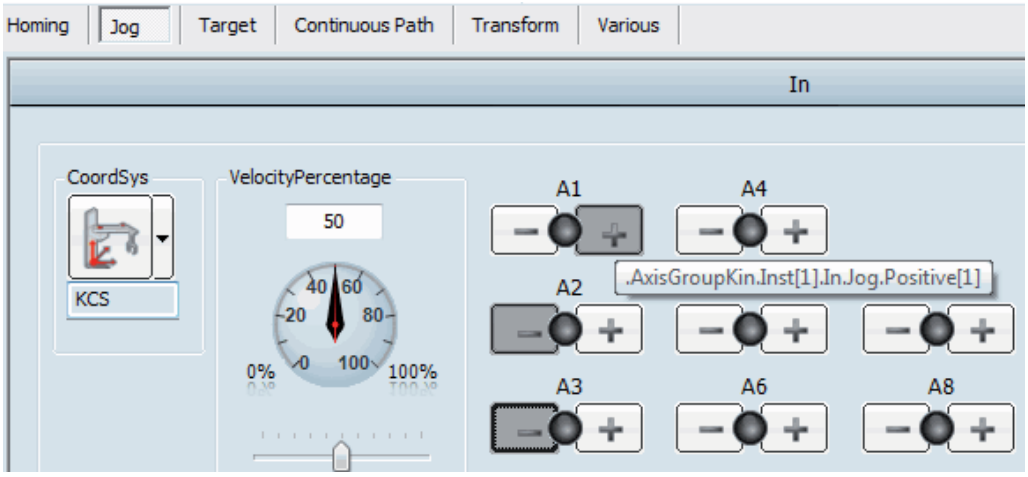
7.1.5 Achsweiser Tippbetrieb

1. Stellen Sie die Betriebsart **KIN_JOG_AXIS** ein.
2. Wechseln Sie zur Registerkarte "Jog" (In-Bereich)
3. Aktivieren Sie für eine oder mehrere Achsen eine Tippbewegung.

   Klicken Sie dazu auf folgende Schaltflächen:

   - **[+]:** Positive Tippbewegung
   - **[-]:** Negative Tippbewegung

20211732 / DE – 07/2014



9308689931

4. Passen Sie, falls erforderlich, die Tippgeschwindigkeit an.

Auswirkung auf das Koordinatensystem

Unabhängig von der Auswahl des Koordinatensystems werden in der Betriebsart **KIN_JOG_AXIS** immer die Achsen im Tippbetrieb verfahren, auch bei Einstellung eines kartesischen Koordinatensystems.

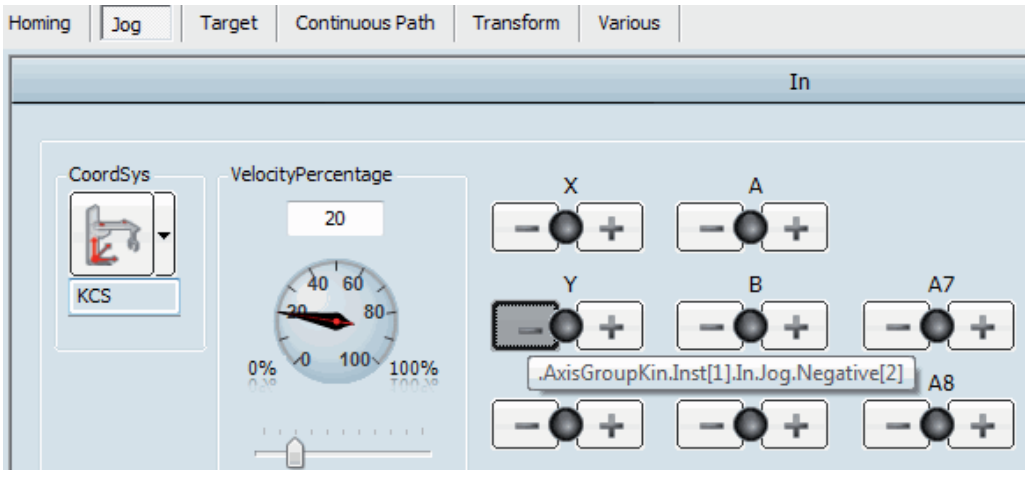
Allerdings wird über die Auswahl des Koordinatensystems das aktuelle Koordinatensystem der Kinematiksteuerung eingestellt, worauf sich auch die Anzeige der aktuellen Position im Out-Bereich bezieht.

7.1.6 Kartesischer Tippbetrieb

1. Stellen Sie die Betriebsart **KIN_JOG_CART** ein.
2. Wechseln Sie zur Registerkarte "Jog" (In-Bereich)
3. Aktivieren Sie für jeden kartesischen Freiheitsgrad des Werkzeugkoordinatensystems (TCS) eine Tippbewegung.

   Klicken Sie dazu auf folgende Schaltflächen:

   - **[+]:** Positive Tippbewegung)
   - **[-]:** Negative Tippbewegung)



9308719115

4. Passen Sie, falls erforderlich, die Tippgeschwindigkeit an.

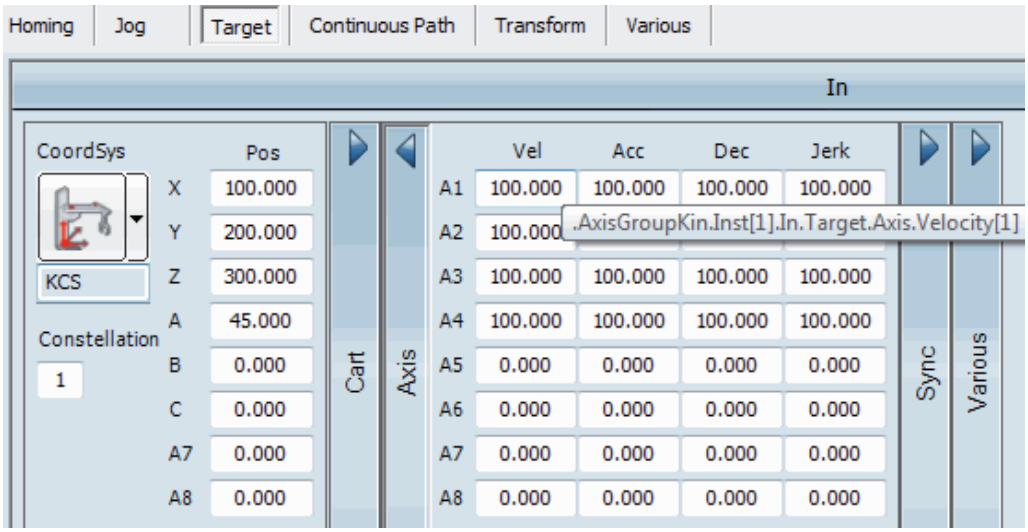
Auswirkung entsprechend dem Koordinatensystem

Die Tippbewegungen erfolgen relativ zum eingestellten Koordinatensystem. Bei Auswahl des Achskoordinatensystems (ACS) werden die Achspositionen im Out-Bereich angezeigt.

Die Tippbewegungen erfolgen bei Auswahl des Achskoordinatensystems (ACS) jedoch in Bezug auf das Kinematikkoordinatensystem (KCS), wie wenn KCS ausgewählt wäre.

7.1.7 Achsweise TARGET-Betriebsart

1. Stellen Sie die Betriebsart **KIN_TARGET_AXIS** ein.
2. Wechseln Sie zur Registerkarte "Target" (In-Bereich)
3. Stellen Sie folgende Bewegungsparameter für die Interpolation der Achsen ein:
   - **Geschwindigkeit (Vel)** in [Anwendereinheiten / sec]
   - **Beschleunigung (Acc)** in [Anwendereinheiten / sec^2]
   - **Bremsbeschleunigung (Dec)** in [Anwendereinheiten / sec^2]
   - **Ruckzeit (Jerk)** in [ms]



9309131019

Auswirkung entsprechend dem Koordinatensystem

Unabhängig von der Auswahl des Koordinatensystems wird der Roboter in Achsenin-terpolation verfahren, also auch bei Einstellung eines kartesischen Koordinatensystems.

Wenn Sie Achskoordinatensystem (ACS) einstellen, geben Sie als Zielposition "Pos" explizite Ziel-Achspositionen ein.

Bei Einstellung eines kartesischen Koordinatensystems geben Sie die kartesischen Zielkoordinaten des Werkzeugkoordinatensystems (TCS) relativ zum eingestellten Koordinatensystem an.

Die Umrechnung in die entsprechenden Achspositionen erfolgt automatisch durch die Kinematiksteuerung.

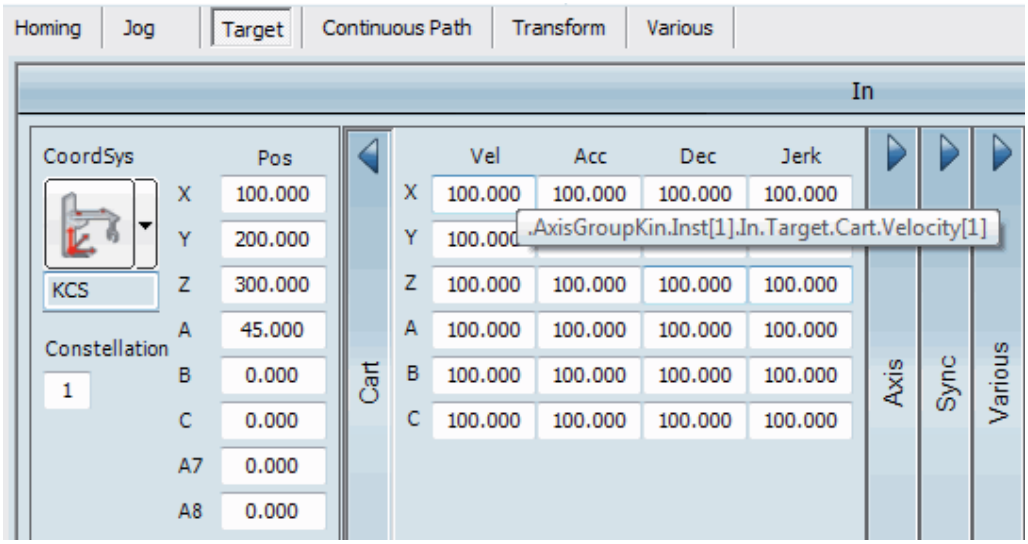
Zusätzlich können Sie in dem Fall auch die "Constellation" einstellen (z. B. 1, 2 für Ellbogen links, rechts bei einer SCARA-Kinematik), in welches die kartesische Zielposition angefahren werden soll. Die Einstellung der "Constellation" 0 bewirkt, dass die aktuelle "Constellation" beibehalten wird.

20211732 / DE – 07/2014

Die aktuelle "Constellation" wird im Out-Bereich rechts oben angezeigt.

7.1.8 Kartesische TARGET-Betriebsart

1. Stellen Sie die Betriebsart **KIN_TARGET_CART** ein.
2. Wechseln Sie zur Registerkarte "Target" (In-Bereich)
3. Stellen Sie folgende Bewegungsparameter für die Interpolation der kartesischen Freiheitsgrade (DOF) ein:
   - **Geschwindigkeit (Vel)** in [Anwendereinheiten / sec]
   - **Beschleunigung (Acc)** in [Anwendereinheiten / sec^2]
   - **Bremsbeschleunigung (Dec)** in [Anwendereinheiten / sec^2]
   - **Ruckzeit (Jerk)** in [ms]



9309404555

Auswirkung entsprechend dem Koordinatensystem

Unabhängig von der Auswahl des Koordinatensystems wird das Werkzeugkoordinatensystem (TCS) kartesisch interpoliert verfahren, also entlang der X-Koordinate, der Y-Koordinate, etc.

Die kartesischen Zielkoordinaten beziehen sich auf das eingestellte Koordinatensystem.

Bei Auswahl des Achskoordinatensystems (ACS) geben Sie die Ziel-Achspositionen ein. Diese werden durch die Kinematiksteuerung automatisch in kartesische Zielkoordinaten umgerechnet und im Kinematikkoordinatensystem (KCS) angefahren, wie wenn KCS eingestellt wäre.

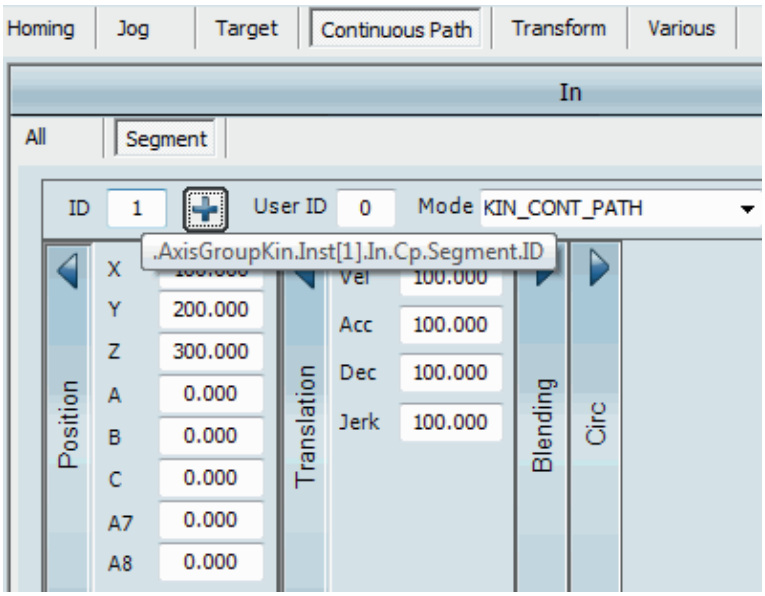
Bei Einstellung von ACS werden die Achspositionen im Out-Bereich angezeigt.

7.1.9 ContinuousPath-Betriebsart

1. Stellen Sie die Betriebsart **KIN_LIN_3D** ein.
2. Wechseln Sie zur Registerkarte "ContinousPath" (In-Bereich)
3. Stellen Sie folgenden Parameter für die translatorische Bewegung entlang der Bahn ein:
   - **Geschwindigkeit (Vel)** in [Anwendereinheiten / sec]

20211732 / DE – 07/2014

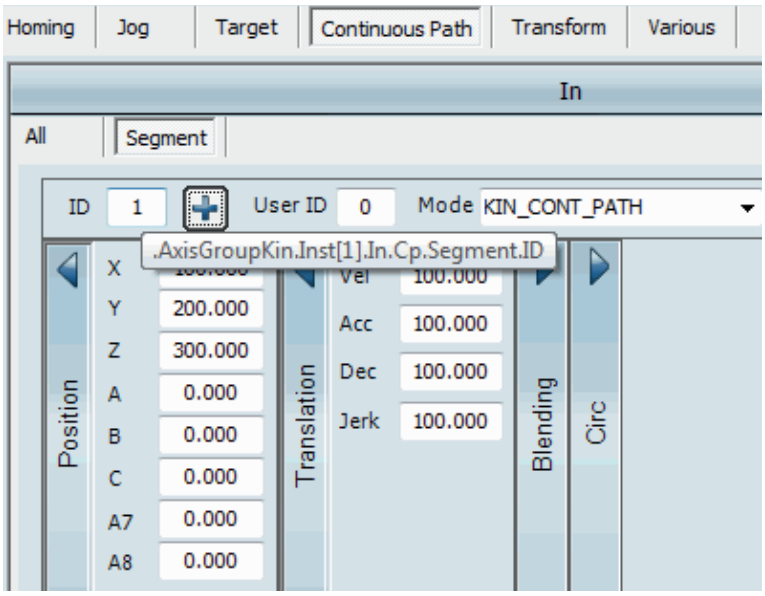
- **Beschleunigung (Acc)** in [Anwendereinheiten / sec^2]
- **Bremsbeschleunigung (Dec)** in [Anwendereinheiten / sec^2]
- **Ruckzeit (Jerk)** in [ms]



9308735115

- Geben Sie die Zielkoordinaten (X, Y etc.) in der Gruppe "Pos" ein.
- Klicken Sie auf die Schaltfläche [+].

  Dadurch erhöhen Sie die Segment-ID um "1" und das aktuelle Segment wird von der Kinematiksteuerung übernommen.



9308735115

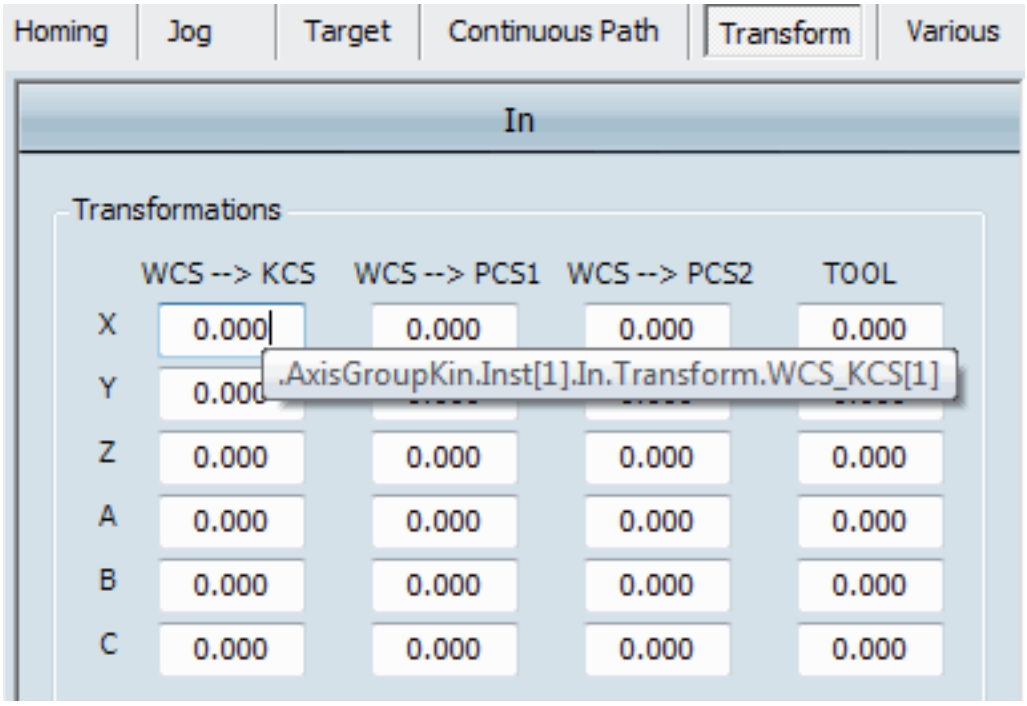
Die Parametrierung der Kreisinterpolation entnehmen Sie dem Kapitel "KIN_CIRC_XY /_YZ / ZX" (→ 🗎 75), sowie dem Kapitel "In.Cp.Segment" (→ 🗎 100).

7.1.10 Wahl der Transformation

1. Wechseln Sie zur Registerkarte "Transform" (In-Bereich)

20211732 / DE – 07/2014

2. Wählen Sie die Transformation aus:
   - **WCS --> KCS**
   - **WCS --> PCS1**
   - **WCS --> PCS2**
   - **TOOL (entspricht FCS --> TCS)**



9309665675

Sprunghafte Änderung

Beachten Sie, dass die sprunghafte Änderung einer Transformation zu einem Schleppfehler oder zu sehr ausgedehnten Bewegungen führen kann. Das ist z. B. der Fall, wenn die Transformation zeitgleich mit der sprunghaften Änderung oder wenige Zyklen der TaskPriority später zur kartesischen Interpolation verwendet wird.

Details hierzu finden Sie im Kapitel "Wechsel des Koordinatensystems" (→ 🗎 127). Die zur Steuerung der Kinematik verwendeten Transformationen dürfen nur sprunghaft geändert werden, wenn eine der folgenden Bedingungen erfüllt ist:

- Die sprunghafte Änderung der Transformation findet statt, während sich die Kinematiksteuerung in Achseninterpolation befindet. Auch in den darauffolgenden 5 Zyklen der TaskPriority muss die Kinematiksteuerung in Achseninterpolation bleiben.

  In Achseninterpolation befindet sich die Kinematiksteuerung in den Betriebsarten **KIN_JOG_AXIS** oder **KIN_TARGET_AXIS** − oder in **KIN_STOP**, wenn zuvor **KIN_JOG_AXIS** oder **KIN_TARGET_AXIS** eingestellt war.
- Wenn die Kinematiksteuerung nicht bereit ist (*Out.General.Kin.Ready* = FALSE).

Sperrung von Transformationen

Für die eineindeutige Zuordnung von Achswerten zu kartesischen Koordinaten sind die zulässigen Transformationen abhängig vom konfigurierten Kinematikmodell teilweise eingeschränkt. Entsprechende Klartextinformationen entnehmen Sie in dem Fall den erscheinenden Messages.

Weiterführende Informationen entnehmen Sie dem Kapitel „Umgang mit Mehrdeutigkeiten der ABC-Orientierungswerte (→ 🗎 138)“.

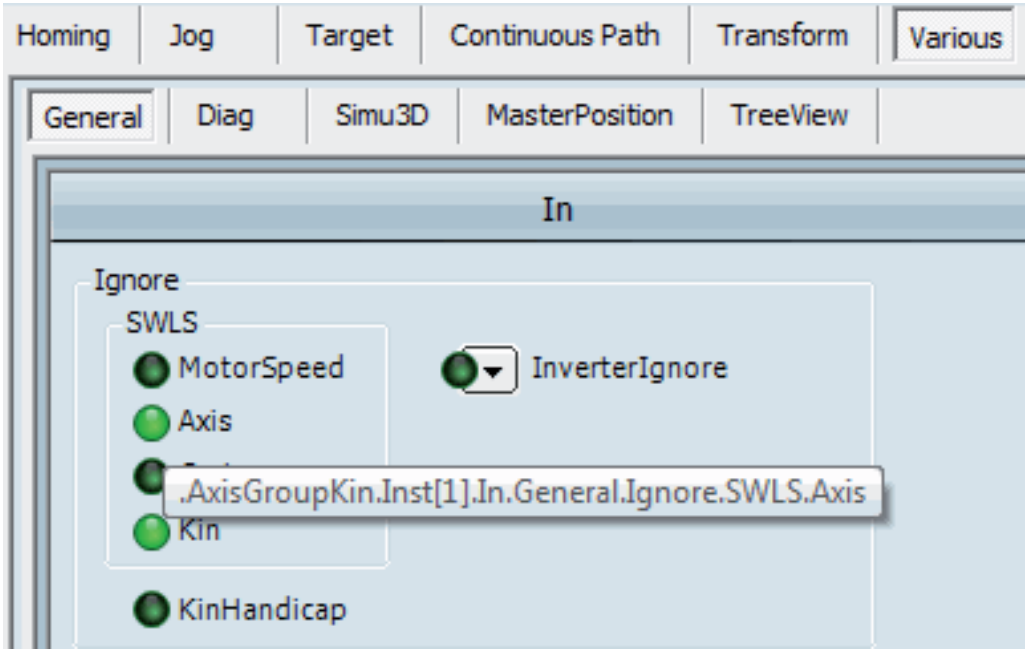
7.1.11 Verschiedene Einstellungen

1. Wechseln Sie zur Registerkarte "Various" (In-Bereich).

   Sie finden hier folgende unterlagerte Registerkarten:

   - **[General]**
   - **[Diag]**
   - **[Simu3D]**
   - **[MasterPosition]**
   - **[TreeView]**

2. Wählen Sie z. B. die Registerkarte "General" und stellen Sie in der Gruppe "Ignore" ein, dass die Einstellungen für die Software-Endschalter ignoriert werden.



9309747979

3. Details zu der Registerkarte "Simu3D" finden Sie im Kapitel "3D-Simulation" (→ 🗎 53).

20211732 / DE – 07/2014

7.2 3D-Simulation

7.2.1 Zweck

Die im MOVITOOLS® MotionStudio integrierte 3D-Simulation ermöglicht die automatische Simulation aller konfigurierten Kinematiken. Dabei passen sich die 3D-Modelle automatisch an die gegebene Konfiguration an.

Die 3D-Simulation ist sehr nützlich für die folgenden Aufgaben:

- Konfigurieren der Kinematik und Anpassen an den realen Roboter
- Erlernen der Bedienung der Kinematiksteuerung
- Überprüfen von Bewegungsabläufen

Die 3D-Simulation kann sowohl offline als auch online ausgeführt werden. Offline bedeutet, dass keine realen Achsen angeschlossen / bewegt werden. Online bedeutet, dass die Simulation parallel zur realen Roboterbewegung ausgeführt wird.

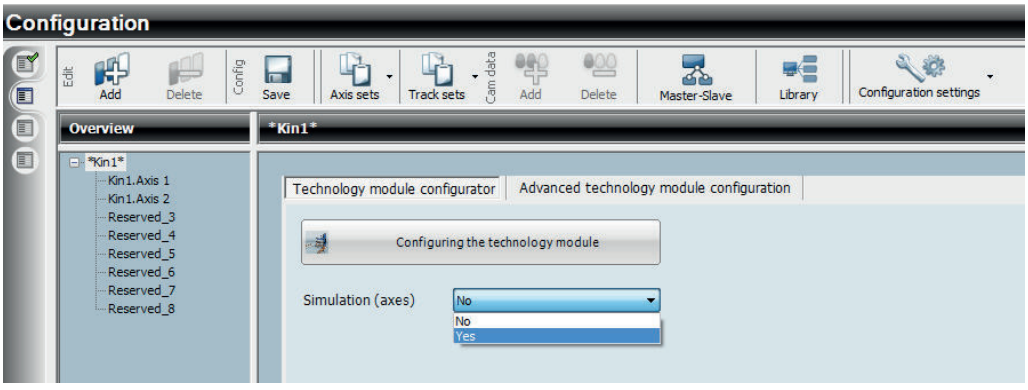
Online ist die 3D-Simulation beispielsweise sehr nützlich, um die Werkzeugbewegung des Roboters zu visualisieren.Die Begutachtung der Werkzeugbahn hinsichtlich Bahnform und Überschleifverhalten ist für das menschliche Auge im Allgemeinen recht schwierig.

7.2.2 Voraussetzungen

Beachten Sie, dass Sie für die Funktion "3D-Simulation" zusätzliche Technologiepunkte benötigen. Details entnehmen Sie dem Kapitel "Erforderliche Technologiepunkte". (→ 🗎 13)

1. Geben Sie im Konfigurationsassistenten die IP-Adresse des Engineering-PC´s an.
2. Wenn Sie keine realen Achsen zur Verfügung haben, aktivieren Sie die Simulation der Achsen im Editor von MultiMotion / MultiMotion light:

   Wechseln Sie hierfür zur Registerkarte "Technologiemodul-Konfigurator" und wählen Sie für Simulation (Achsen) "Ja":



9308228107

3. Stellen Sie eine Ethernet-Verbindung zwischen der MOVI-PLC® und dem Engineering-PC her.

   Hinweis: Die 3D-Simulation ist über USB oder Feldbus nicht möglich!

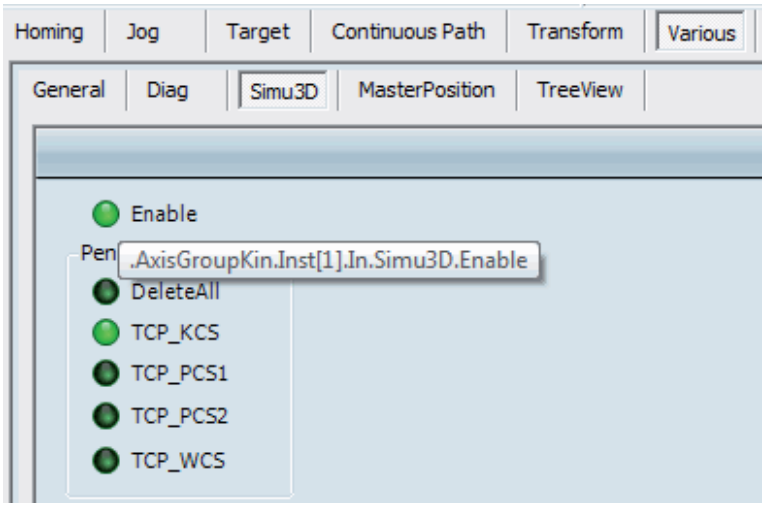
   Die Kommunikation mit der 3D-Simulation geschieht ausschließlich über die Ethernet-Engineering-Schnittstelle (X37 bei MOVI-PLC® advanced, LAN 3 bei MOVI-PLC® power).

Damit Sie die 3D-Simulation ausführen können, muss die Variable *.AxisGroup-Kin.Inst[..].In.Simu3D.Enable* = TRUE sein. Die Standardbelegung ist TRUE

Sie können die Variable im Anwenderprogramm aktivieren oder im Kinematikmonitor:

20211732 / DE – 07/2014

4. Wechseln Sie zu der Registerkarte "Various" / "Simu3D".



9308210955

5. Durch Klicken auf die LED-Symbole können Sie Stifte am TCP aktivieren, welche Linien in Bezug auf verschiedene Koordinatensysteme zeichnen (z. B. TCP_KCS). Ebenso können Sie die gezeichneten Linien auch wieder löschen (DeleteAll).

7.2.3 Starten der 3D-Simulation

Die 3D-Simulation startet automatisch, wenn Sie mit dem Kinematikmonitor arbeiten und *In.Simu3D.Enable* = TRUE ist.

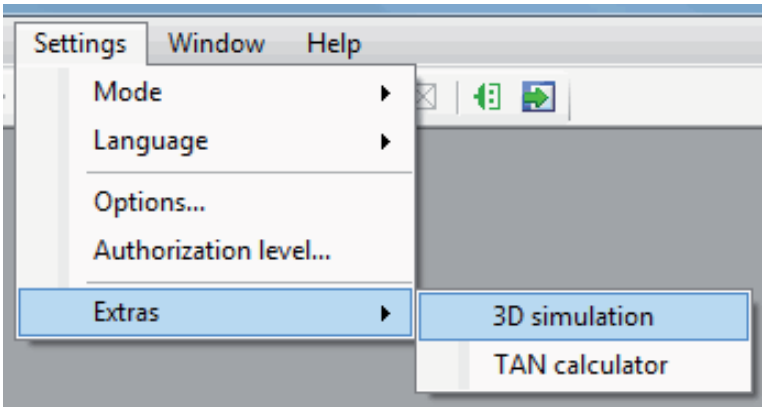
Sie können die 3D-Simulation auch explizit auf eine der folgenden Arten starten:

- Im Kinematikmonitor:

  Klicken Sie in der Registerkarte "Various" / "Simu3D" auf die Schaltfläche [Start Simu3D Tool].

- Mit MOVITOOLS® MotionStudio:

  Starten Sie die 3D-Simulation von MOVITOOLS® MotionStudio mit dem Menübefehl [Einstellungen] > [Extras] > [3D-Simulation].



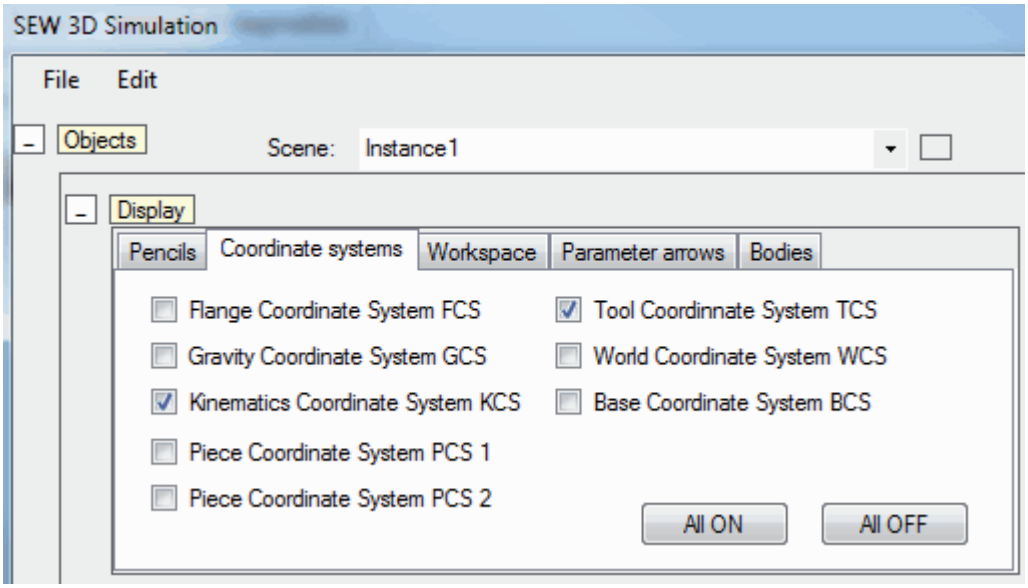
9308214667

7.2.4 Anpassen der 3D-Simulation

Sie können das Erscheinungsbild der 3D-Simulation in dem Kontrollfenster anpassen, das mit der 3D-Simulation automatisch gestartet wird.

20211732 / DE – 07/2014

1. Starten Sie die 3D-Simulation, wie unter "Starten der 3D-Simulation" (→ 🗎 54) beschrieben.



9308207755

2. Passen Sie die 3D-Simulation mithilfe der folgenden Registerkarten an:

- **"Pencils":** Hier stellen Sie z. B. die Farbe der Linien ein. Die Stifte können hier jedoch nicht ein-/ausgeschaltet werden.
- **"Koordinatensystem":** Blenden Sie hier verschiedene Koordinatensysteme ein, um die aktuell gültigen Transformationen überprüfen zu können.
- **"Workspace":** Blenden Sie hier Pfeile ein zur Anzeige der Nullpunkte und Richtungen der gültigen Verfahrbereiche für die Achsen, die kartesischen Dimensionen sowie die kinematischen Limitierungen.
- **"Parameter arrows":** Lassen Sie sich hier nützliche Informationen durch Pfeile anzeigen (z. B. zur Kennzeichnung der Kinematikparameter wie Armlängen und Versätze oder Pfeile zur Anzeige der Software-Endschalter von Achsen).
- **"Bodies":** Ausgewählte Körper lassen sich ein-/ausblenden

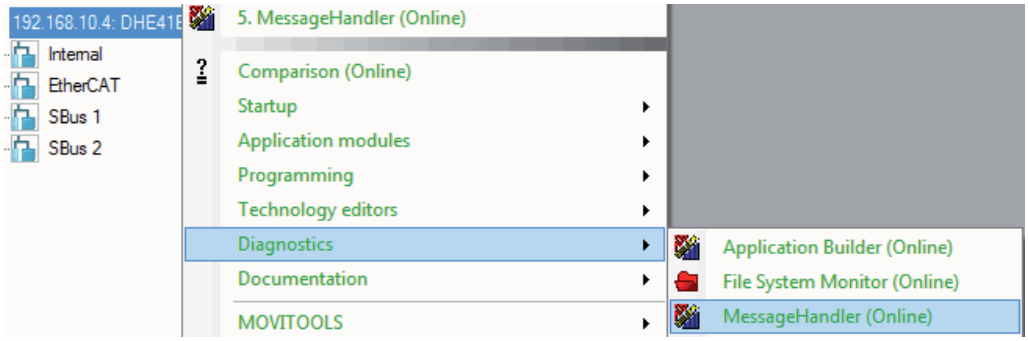
7.3 MessageHandler

Der MessageHandler ist ein Diagnosewerkzeug, das Ihnen in übersichtlicher Form Klartextinformationen zu Fehlern anzeigt. Diese ergänzen die Information, die sie zu der Fehlernummer (Error-ID) erhalten und erleichtert Ihnen die Fehlerbehebung.

Gehen Sie folgendermaßen vor, um den "MessageHandler" zu starten:

1. Starten Sie die Engineering-Software "MOVITOOLS® MotionStudio".
2. Markieren Sie in der Netzwerksicht das Symbol für den Controller.

20211732 / DE – 07/2014



9308613003

3. Öffnen Sie mit der rechten Maustaste das Kontextmenü und wählen Sie den Menübefehl [Diagnose] > [MessageHandler].

   Der "MessageHandler" öffnet sich.

   Aktuelle Fehlermeldungen finden Sie im oberen Bereich. Bereits archivierte Fehlermeldungen werden darunter angezeigt.

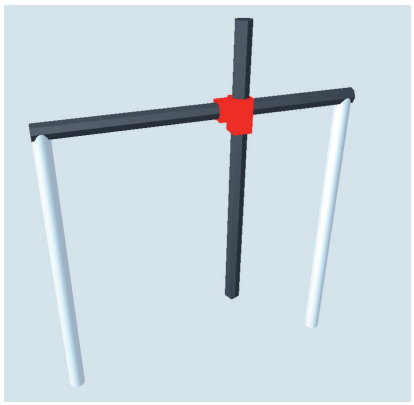
20211732 / DE – 07/2014

8 Kinematikmodelle

Dieses Kapitel gibt Ihnen eine Überblick über die Kinematikmodelle, die Sie im Konfigurationsassistenten auswählen können.

8.1 CARTESIAN GANTRY

CARTESIAN_GANTRY _LL_XY_M10

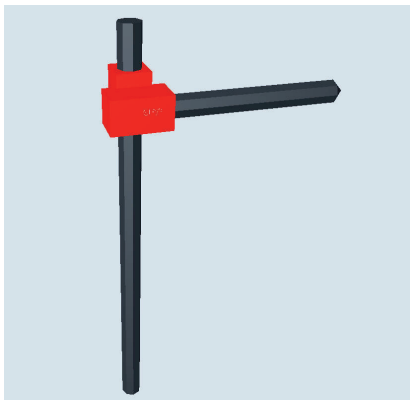


9308233355

Kartesisches Gantry (Portal) mit 2 linearen Achsen:

- Achse 1 entspricht der X-Richtung
- Achse 2 der Y-Richtung.

CARTESIAN_GANTRY _LL_ZX_M10



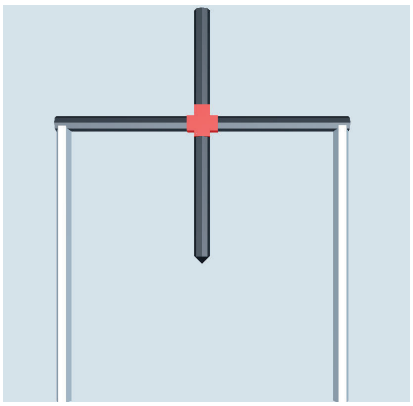
9308238987

Kartesisches Gantry (Portal) mit 2 linearen Achsen:

- Achse 1 entspricht der Z-Richtung
- Achse 2 der X-Richtung.

Mithilfe eines Parameters kann der Winkel zwischen Z- und X-Richtung auch auf Werte ungleich 90° eingestellt werden.

CARTESIAN_GANTRY _LL_ZX_M15



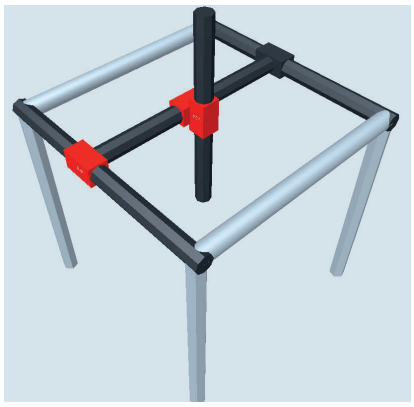
9308245003

Kartesisches Gantry (Portal) mit 2 linearen Achsen:

- Achse 1 entspricht der X-Richtung
- Achse 2 der Z-Richtung

20211732 / DE – 07/2014

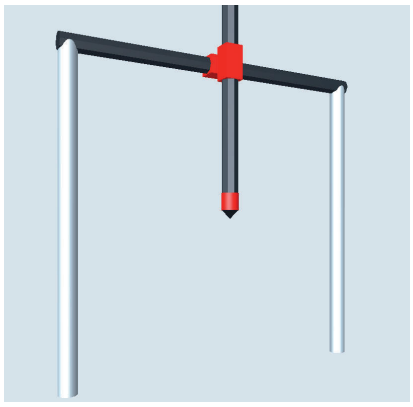
CARTESIAN_GANTRY _LLL_XYZ_M10



9308281483

Kartesisches Gantry (Portal) mit 3 linearen Achsen.

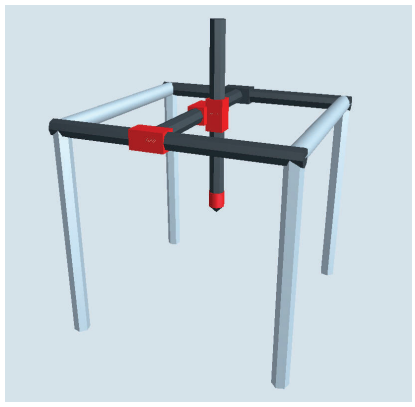
CARTESIAN_GANTRY _LLR_XYB_M10



9308312715

Kartesisches Gantry (Portal) mit 2 linearen Achsen (X, Y) und 1 rotativen Achse (Drehung um Y).

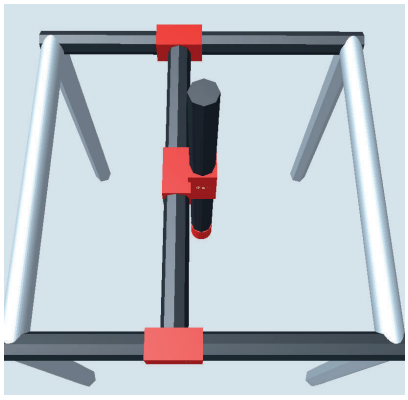
CARTESIAN_GANTRY _LLLR_XYZA_M10



9308272907

Kartesisches Gantry (Portal) mit 3 linearen und einer rotativen Achse (Drehung um Z).

CARTESIAN_GANTRY _LLLLR_XYZA_M10

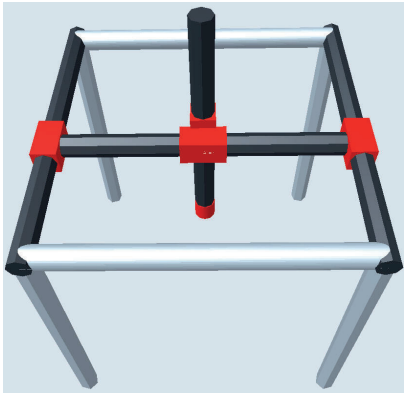


9308251403

Kartesisches Gantry (Portal) mit 4 linearen Achsen und 1 rotativen Achse (Drehung um Z).

Die X-Richtung wird mit 2 Achsen (A1, A2) realisiert.

CARTESIAN_GANTRY _LLLLR_XYZA_M20

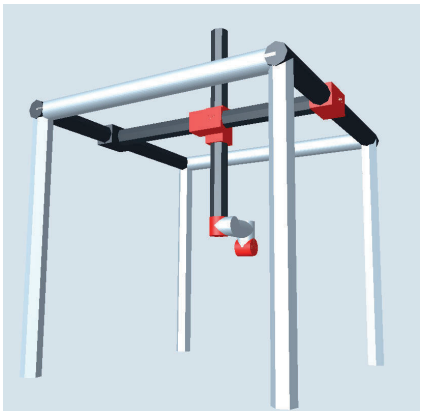


9308258187

Kartesisches Gantry (Portal) mit 4 linearen Achsen und 1 rotativen Achse (Drehung um Z).

Die Y-Richtung wird mit 2 Achsen (A2, A3) realisiert.

CARTESIAN_GANTRY _LLLRR_XYZAB_M10

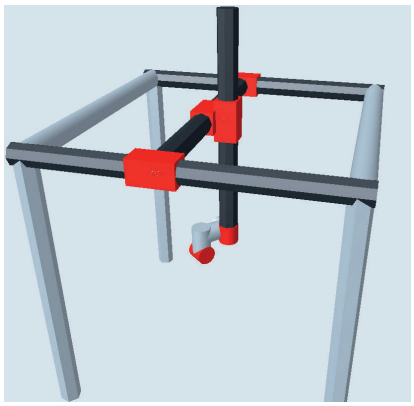


9308303627

Kartesisches Gantry (Portal) mit 3 linearen Achsen und 2 rotativen Achsen.

20211732 / DE – 07/2014

CARTESIAN_GANTRY_LLLLRR_XYZAB_M10



9308265355

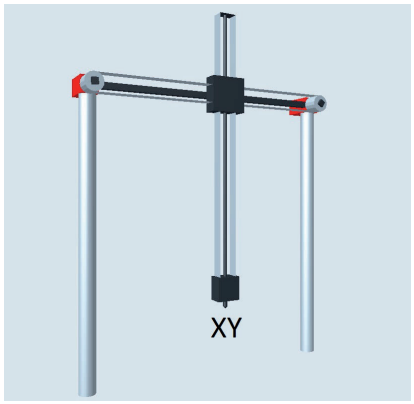
Kartesisches Gantry (Portal) mit 4 linearen und 2 rotativen Achsen.

Die X-Richtung wird mit 2 linearen Achsen (A1, A2) realisiert.

20211732 / DE – 07/2014

8.2 ROLLER GANTRY

ROLLER_GANTRY_LL_XY_M10

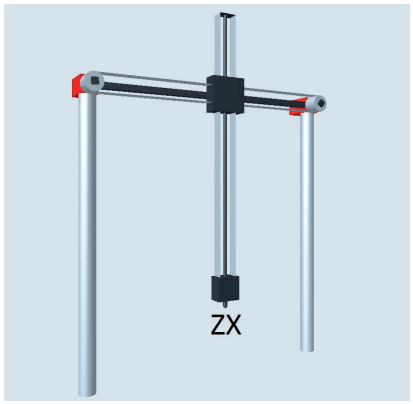


9310212747

Roller Gantry mit 2 Achsen.

Die Bewegungen erfolgen in der XY-Ebene.

ROLLER_GANTRY_LL_ZX_M10

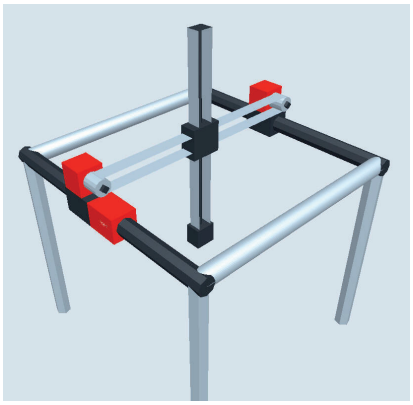


9310234123

Roller Gantry mit 2 Achsen.

Die Bewegungen erfolgen in der ZX-Ebene.

ROLLER_GANTRY _LLL_XYZ_M10

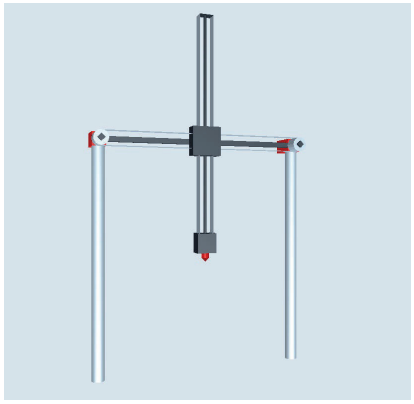


9310255883

Roller Gantry mit 3 Achsen:

- Achse 1 entspricht der X-Richtung
- Achse 2 und 3 treiben die Roller-Riemen in der YZ-Ebene an.

ROLLER_GANTRY _LLR_XYB_M10

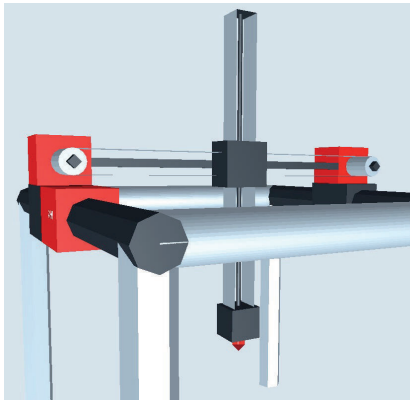


9310370443

Roller Gantry mit 3 Achsen:

- Die Achsen 1 und 2 treiben die Roller-Riemen in der XY-Ebene an.
- Achse 3 entspricht der Drehung B.

ROLLER_GANTRY _LLLR_XY-ZA_M10

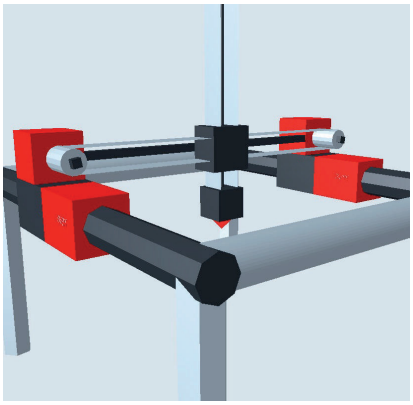


9310323467

Roller Gantry mit 4 Achsen:

- Achse 1 entspricht der X-Richtung.
- Die Achsen 2 und 3 treiben die Roller-Riemen in der YZ-Ebene an.
- Achse 4 entspricht der Drehung A.

ROLLER_GANTRY _LLLLR_XY-ZA_M10



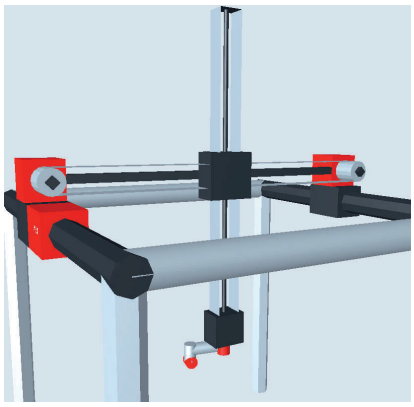
9310278027

Roller Gantry mit 5 Achsen:

- Die X-Richtung wird mit 2 Achsen (A1, A2) realisiert.
- Die Achsen 3 und 4 treiben die Roller-Riemen in der YZ-Ebene an.
- Achse 5 entspricht der Drehung A.

20211732 / DE – 07/2014

ROLLER_GANTRY _LLLRR_XYZAB_M10

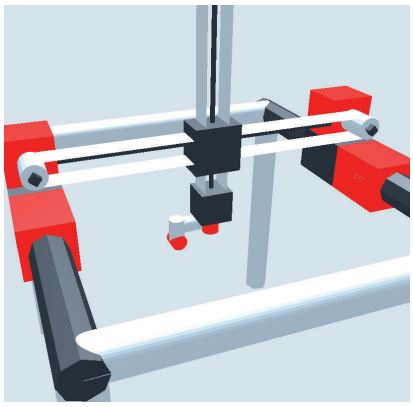


9310346763

Roller Gantry mit 5 Achsen:

- Achse 1 entspricht der X-Richtung.
- Die Achsen 2 und 3 treiben die Roller-Riemen in der YZ-Ebene an.
- Achse 4 entspricht der Drehung A.
- Achse 5 entspricht der Drehung B.

ROLLER_GANTRY _LLLLRR_XYZAB_M10



9310300555

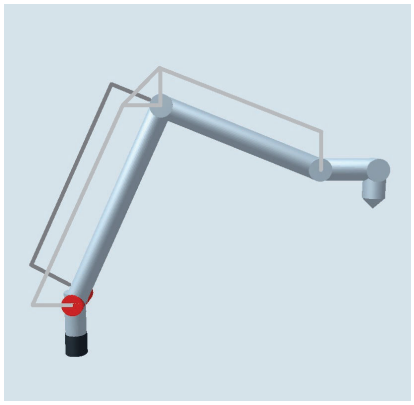
Roller Gantry mit 6 Achsen:

- Die X-Richtung wird mit 2 Achsen (A1, A2) realisiert.
- Die Achsen 3 und 4 treiben die Roller-Riemen in der YZ-Ebene an.
- Achse 5 entspricht der Drehung A.
- Achse 6 entspricht der Drehung B.

20211732 / DE – 07/2014

8.3 SCARA

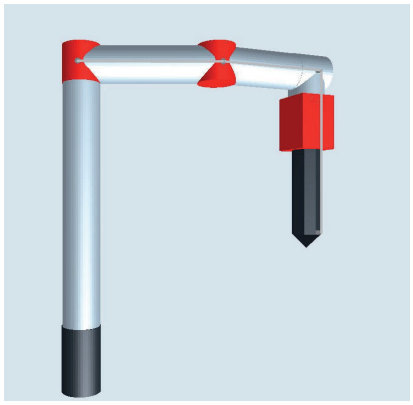
SCARA_RR_XY_M20



9310418955

SCARA-Kinematik mit 2 rotativen Achsen. Der Unterarm wird mithilfe eines Parallelogramms oder Riemens durch den stationären Antrieb 2 bewegt.

SCARA_RRL_XYZ_M20



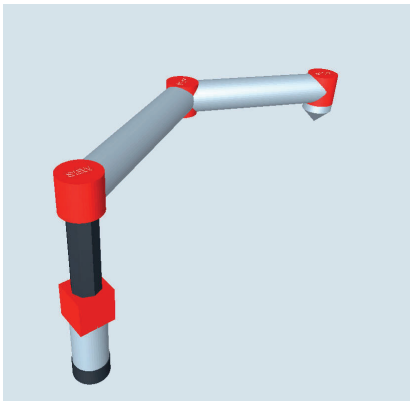
9310443787

SCARA-Kinematik mit 2 rotativen Achsen (Schulter, Ellbogen) und 1 linearen Achse. Das Flanschkoordinatensystem (FCS) kann nur translatorisch bewegt, also nicht gedreht werden.

Die Flanschorientierung wird mithilfe eines Parallelogramms oder einen Riemen konstant gehalten - oder das Werkzeug besteht z. B. nur aus einem Stift, dessen Orientierung keine Rolle spielt.

Die lineare Achse 3 entspricht der Z-Richtung.

SCARA_LRRR_XYZA_M10

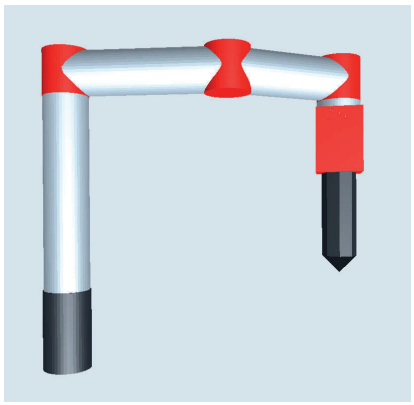


9310394507

SCARA-Kinematik mit 1 linearen Achse und 3 rotativen Achsen (Schulter, Ellbogen, Handgelenk).

20211732 / DE – 07/2014

SCARA_RRRL_XYZA_M10

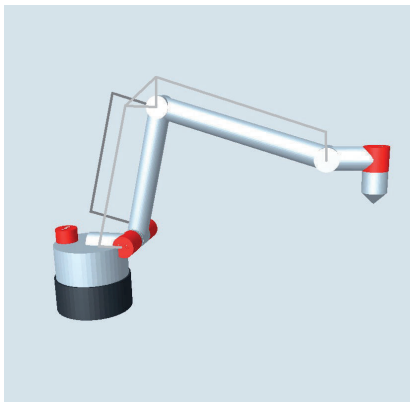


9310469003

SCARA-Kinematik mit 4 Achsen:

- 3 rotativen Achsen (Schulter, Ellbogen, Handgelenk)
- 1 lineare Achse

SCARA_RRRR_XYZA_M60



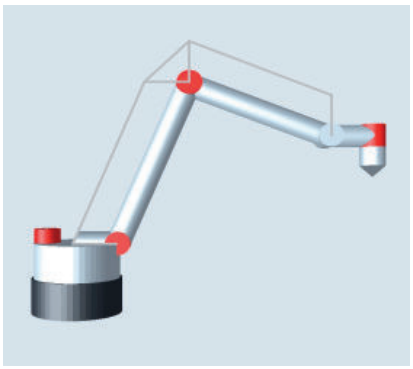
9310494603

SCARA-Kinematik mit 4 rotativen Achsen

(Drehturm, Schulter, Ellbogen, Handgelenk).

Der Unterarm wird mithilfe eines Parallelogramms oder Riemens durch den Antrieb 3 bewegt, der auf dem Drehturm angebracht ist.

SCARA_RRRR_XYZA_M65



9637703179

SCARA mit 4 rotativen Achsen

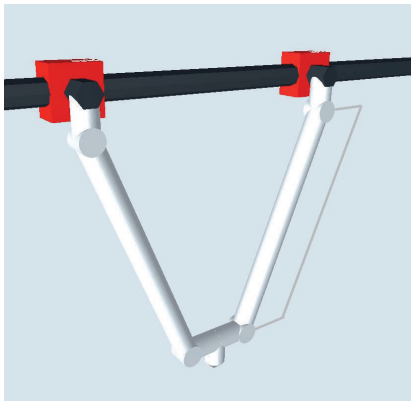
(Drehturm, Schulter, Ellbogen, Handgelenk).

Jedes Drehgelenk wird direkt durch einen Antrieb bewegt.

20211732 / DE – 07/2014

8.4 DELTA

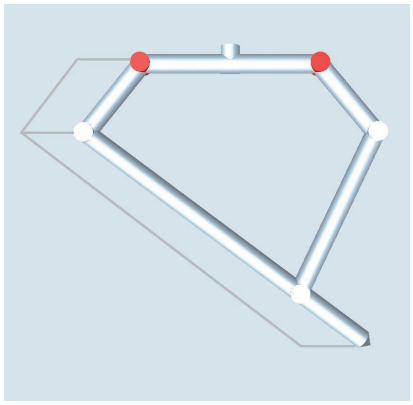
DELTA_LL_XY_M10



9308455051

Delta-Kinematik mit 2 linearen Achsen.

DELTA_RR_XY_M20



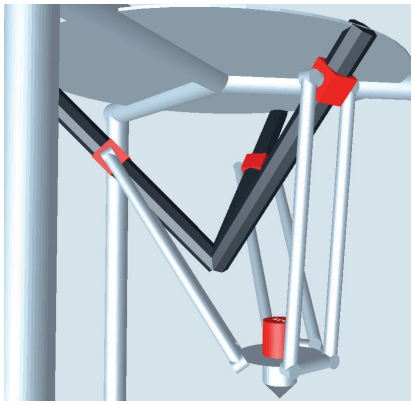
9308465675

Delta-Kinematik mit 2 rotativen Achsen.

20211732 / DE – 07/2014

8.5 TRIPOD

TRIPOD_LLLR_XYZA_M10



12852397067

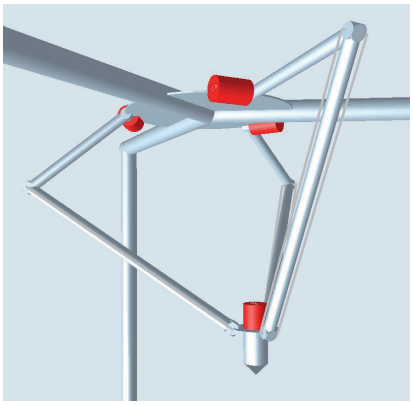
Tripod mit 3 Linearachsen und 1 rotativen Achse.

TRIPOD_RRR_XYZ_M10

9310520587

Tripod mit 3 rotativen Achsen.

TRIPOD_RRRR_XYZA_M10



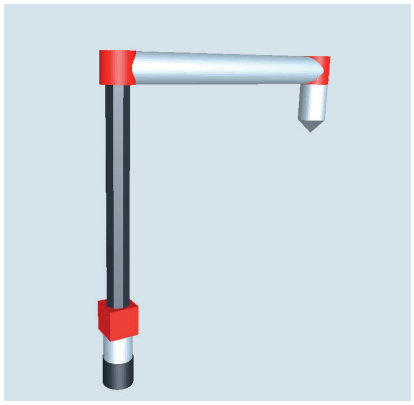
9310546955

Tripod mit 4 rotativen Achsen. Achse 4 dreht das Werkzeug um die Z-Achse.

20211732 / DE – 07/2014

8.6 MIXED

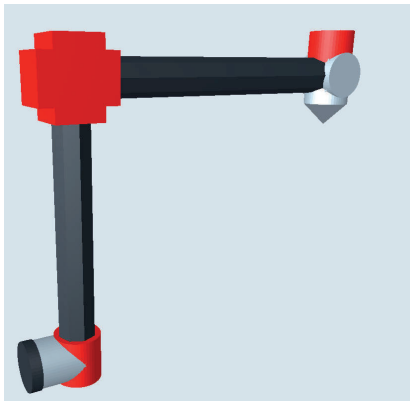
MIXED_LRR_ZXA_M10



9308658315

Kinematik mit 1 Linear- und 2 rotativen Achsen, die Bewegungen des Flansches auf der Oberfläche eines Zylinders um die Linearachse (vertikal Z, horizontal X) sowie Drehungen des Flansches (A) ermöglicht.

MIXED_RLLR_XYZA_M10



9308673931

Kinematik mit 1 rotativen Achse im Sockel, 2 Linearachsen und 1 rotativen Achse am Flansch.

8.7 USER-Kinematik

Weitere Kinematikmodelle (z. B. Quadropod, Hexapod, Articulated, None etc.) bis hin zu Spezialkonstruktionen) lassen sich als USER-Kinematik einbinden. Details hierzu entnehmen Sie dem Kapitel "Einbinden einer USER-Kinematik" (→ 🗎 138).

Wenn Sie darüber hinaus Informationen zu speziellen Kinematikmodellen benötigen, nehmen Sie Kontakt zu SEW-EURODRIVE auf.

20211732 / DE – 07/2014

9 Interpolierende Betriebsarten

In diesem Kapitel erhalten Sie zunächst einen Überblick über alle interpolierenden Betriebsarten. Danach werden diese Betriebsarten detailliert behandelt und die Betriebsarten für das Überschleifen bei TARGET und ContinuousPath verglichen.

Zum Schluss erfahren Sie die Einsatzmöglichkeiten und Vorteile, die eine Kombination beider Betriebsarten bietet.

9.1 Überblick

Die folgende Aufstellung zeigt ihnen die Betriebsarten und ihre Anwendung:

Tippbetriebsarten	Wenn Sie Achsen innerhalb von kartesischen Freiheitsgraden (DOF) bewegen möchten. Voraussetzung ist, dass das betreffende Steuersignal (Befehls-Bit) aktiv ist.
TARGET-Positionierung	Wenn Sie eine Zielposition achsweise oder kartesisch anfahren möchten.
ContinuousPath-Interpolation	Wenn Sie eine Folge von Bahnsegmenten, (die sich im Allgemeinen geometrisch im Raum befinden), abfahren.

Die folgende Tabelle gibt Ihnen einen Überblick über die Funktionen und Merkmale dieser Betriebsarten:

Betriebsarten		**Funktionen**	**Merkmale**
Tippbetriebsarten	KIN_JOG_AXIS	Tippen der Einzelachsen (Kinematik- und Hilfsachsen)	
	KIN_JOG_CART	Tippen des TCP entlang der Koordinatenachsen (X, Y, Z, A, B, C) im aktuell eingestellten Koordinatensystem.Gleichzeitig können Hilfsachsen achsweise getippt werden.	
TARGET-Positionierung	KIN_TARGET _AXIS (→ 🗎 71)	Positionieren der Einzelachsen	• Einzelachsen bzw. kartesische Freiheitsgrade synchronisierbar • Zielposition kann in beiden Betriebsarten achsweise oder kartesisch spezifiziert werden. (Details hierzu finden Sie im Kapitel "Wechsel des Koordinatensystems".) (→ 🗎 127)
	KIN_TARGET_ _CART (→ 🗎 72)	Positionieren der kartesischen Freiheitsgrade des Werkzeugmittelpunkts (TCP), z. B. X, Y, Z	

20211732 / DE – 07/2014

Betriebsarten		Funktionen	Merkmale
ContinuousPath-Interpolation	KIN_LIN_XY/ _YZ/ _ZX (→ 🗎 75)	Bewegung des Werkzeugmittelpunkts (TCP) entlang einer Geraden in einer Hauptebene.	• Bahnsegmente werden durch die Kinematiksteuerung übernommen und in eine "Queue" (Auftragsliste) eingetragen. • Das in den Segmenten parametrierte Überschleifen wird durch die Kinematiksteuerung automatisch durchgeführt.
	KIN_LIN_3D (→ 🗎 75)	Bewegung des Werkzeugmittelpunkts (TCP) entlang einer Geraden im 3D-Raum.	
	KIN_CIRC_XY/ _YZ/ _ZX (→ 🗎 75)	Bewegung des Werkzeugmittelpunkts (TCP) entlang eines Kreises.	

9.2 Tippbetrieb

9.2.1 Vorgabe der Richtung und kinematischer Bewegungsgrößen

Richtungsvorgabe: Im Tippbetrieb erfolgt die Richtungsvorgabe in den Eingangssignalen *AxisGroupKin.Inst[..].In.Jog.Positive* und *AxisGroupKin.Inst[..].In.Jog.Negative* durch Setzen der entsprechenden Array-Elemente.

Ist das gleiche Array-Element in beiden Variablen gesetzt, wird nicht verfahren.

Kinematische Bewegungsgrößen: Die gewünschte Tippgeschwindigkeit kann zyklisch mit dem Eingangssignal *AxisGroupKin.Inst[..].In.Jog.VelocityPercentage* variiert werden. Ist der Prozentsatz auf den Wert "100" eingestellt, verfährt die Steuerung soweit möglich mit der konfigurierten Geschwindigkeit zu verfahren.

9.3 TARGET-Betriebsarten

9.3.1 Prinzip

Bei den TARGET-Betriebsarten übernimmt die Bahnsteuerung zyklisch neue Zielvorgaben (TARGET-Interpolation).

Die Betriebsarten **KIN_TARGET_AXIS** und **KIN_TARGET_CART** bestimmen, wie zum Zielpunkt verfahren wird, d. h. ob im Achsraum oder im kartesischen Raum interpoliert wird. Es geht hierbei also um die Art der Bewegung.

Der Zielpunkt kann jedoch in beiden Betriebsarten mittels Achswerten (ACS) oder mittels kartesischer Koordinaten (z. B. KCS) vorgegeben werden. Die Kinematiksteuerung rechnet die Zielkoordinaten bei Bedarf automatisch in Ziel-Achswerte um (bei **KIN_TARGET_AXIS**) bzw. in kartesische Ziel-Koordinaten (bei **KIN_TARGET_CART**).

Eine detaillierte Übersicht der möglichen Kombinationen von Betriebsart und Zielvorgabe finden Sie im Kapitel "Wechsel des Koordinatensystems" (→ 🗎 127).

Die Programmierung der TARGET-Betriebsarten ähnelt der üblichen Programmierung von Einzelachsen. Daher findet sich der Anwender von Bewegungssteuerungen einzelner Achsen (z. B. mit MultiMotion) direkt zurecht und er kann komplexe Kinematiken bis hin zum 6-Achs-Knickarmroboter genauso einfach programmieren wie Einzelachsen.

Die Zielwerte werden in das folgende Eingangssignal eingetragen:

- *AxisGroupKin.Inst[..].In.Target.Position*

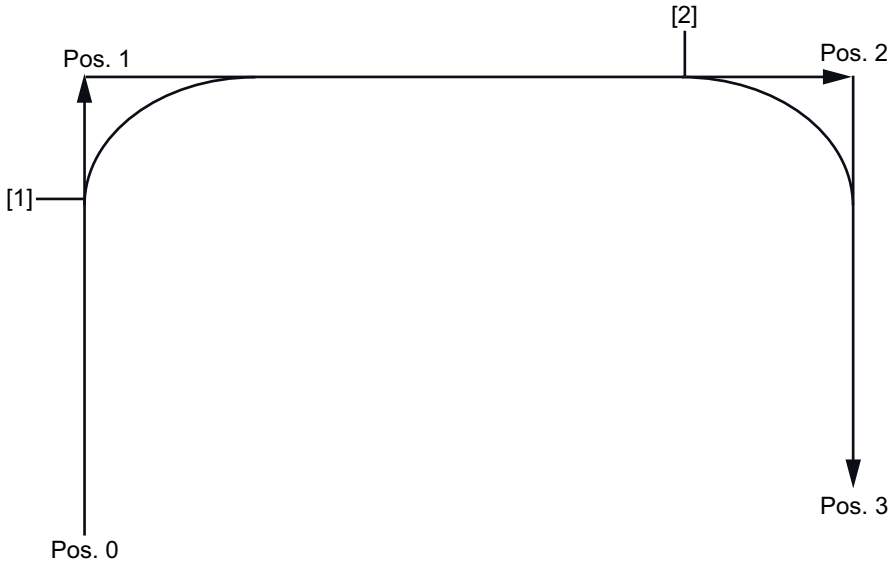
20211732 / DE – 07/2014

Die kinematischen Bewegungsgrößen werden in folgenden Eingangssignalen eingetragen:

- *AxisGroupKin.Inst[..].In.Target.Axis.*
- *AxisGroupKin.Inst[..].In.Target.Cart.*

9.3.2 Beispiel: Pick-and-Place

Die folgende Darstellung zeigt das Überschleifen von Zielen (TARGET-Überschleifen) im Bahnverlauf einer Pick-and-Place-Anwendung:



[1] Umschaltung des Ziels auf Position 2

[2] Umschaltung des Ziels auf Position 3

Ablauf: Die Eingangsvariable *AxisGroupKin.Inst[n].In.Target.Position* wird nacheinander mit den Zielwerten (Positionen) belegt:

1. Zu Beginn entspricht der Zielwert der Position 1.
2. In der Nähe von Position 1 angekommen, wird der Zielwert auf Position 2 umgeschaltet.
3. In der Nähe von Position 2 angekommen, wird der Zielwert auf Position 3 umgeschaltet.

Das Überschleifen von Zielen (TARGET-Überschleifen) wird im Anwenderprogramm initiiert.

Auf diese Weise werden üblicherweise kartesische Portale für Handling-Aufgaben programmiert.

In der Betriebsart **KIN_TARGET_CART** erfolgt die Bewegung nicht achsweise. Stattdessen werden die kartesischen Koordinaten (X/Y/Z/A/B/C) des Werkzeugmittelpunkts (TCP), entlang der kartesischen Achsen (X/Y/ ...) zunächst zur Position 1 verfahren, dann zu Position 2 usw.

Dabei ist es unerheblich um welche Art von Kinematik es sich handelt (Portalkinematik, SCARA, Knickarmroboter etc.).

Die Berechnung und Steuerung der erforderlichen Achsverläufe übernimmt die Kinematiksteuerung automatisch. Die Steuerung der vom Anwender programmierten Positionierung der X/Y/Z/…-Koordinaten des Werkzeugkoordinatensystems (TCP) erfolgt im 3D-/6D-Raum.

20211732 / DE – 07/2014

9.3.3 Vorteile

Bei der üblichen Programmierung eines kartesischen Portals werden die kinematischen Bewegungsgrößen zum Verfahren der Achsen in den Umrichtern eingestellt. Das heißt, dass sich die Bahnen durch gleichzeitiges Verfahren voneinander unabhängiger, jedoch zeitlich koordinierter Achsen ergeben.

Genauso werden bei der TARGET-Interpolation für jede Dimension (Achsen bei **KIN_TARGET_AXIS**, kartesisch bei **KIN_TARGET_CART**) voneinander getrennte kinematische Bewegungsgrößen eingestellt (Eingangssignale *AxisGroupKin.Inst[..].In.Target.Axis* / *AxisGroupKin.Inst[..].In.Target.Cart*).

Die Bahn des Werkzeugmittelpunkts (TCP) ergibt sich durch gleichzeitiges Steuern der Profilgeneratoren für jede beteiligte Dimension zu den aktuell eingetragenen Zielkoordinaten. Auf diese Art lassen sich harmonische und schnelle Bahnen erzeugen, die z. B. für Handling-Aufgaben sehr gut geeignet sind. Der Anwender kann für jede Richtung der Bewegung explizit die maximal zulässigen kinematischen Bewegungsgrößen einstellen.

Folgende Vorteile ergeben sich bei TARGET-Interpolation gegenüber üblicher Programmierung kartesischer Portale:

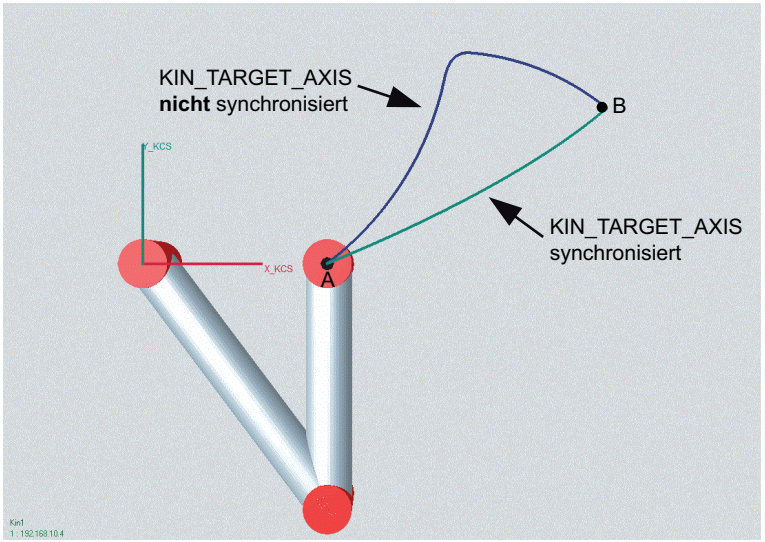
- Gleiche gewohnte Bahnprogrammierung für kartesische Portale und komplexe Kinematiken.
- Möglichkeit der Synchronisation der an der Bewegung beteiligten Dimensionen, so dass alle Komponenten im Allgemeinen gleichzeitig das Ziel erreichen. Bei Rast-in-Rast-Bewegungen können sich so bei einheitlicher Einstellung der Bewegungsparameter für die Bewegungsgrößen auch annähernd geradlinige Bewegungen ergeben (Eingangssignal *AxisGroupKin.Inst[..].In.Target.Synchronization)*.

20211732 / DE – 07/2014

9.3.4 KIN_TARGET_AXIS

Die Betriebsart **KIN_TARGET_AXIS** ermöglicht das achsweise Verfahren zu Zielachswerten. Dabei werden jeder Achse eigene kinematische Bewegungsgrößen zugeordnet.

Die folgende Abbildung zeigt 2 mögliche Bahnverläufe des Werkzeugmittelpunkts (TCP) bei Bewegung einer SCARA-Kinematik zwischen den Punkten A und B:



9308521483

Beim oberen Bogen sind die Achsprofile nicht synchronisiert.

Beim unteren, gleichmäßigeren Bogen sind die Achsprofile synchronisiert (*AxisGroupKin.Inst[..].In.Target.Synchronization[0]* = TRUE).

Bei diesem Kinematikmodell ergibt sich trotz Synchronisation im Raum keine Gerade. Jede Achse verfährt zu den Zielachswerten, um den Zielpunkt im Raum zu erreichen. Bei eingestellter Synchronisation erreichen alle Achsen die Zielachswerte gleichzeitig.

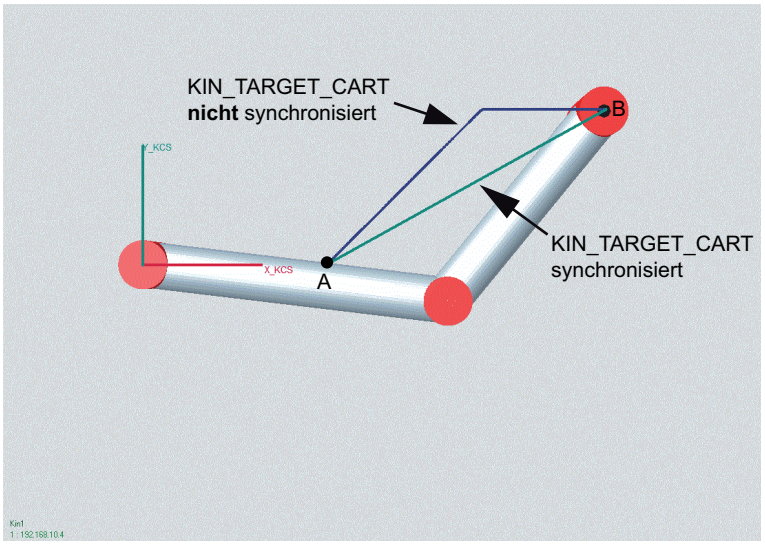
20211732 / DE – 07/2014

9.3.5 KIN_TARGET_CART

Die Betriebsart **KIN_TARGET_CART** ermöglicht das Verfahren des Werkzeugmittelpunkts (TCS) entlang der Koordinatenachsen (X, Y, Z, A, B, C) im aktuell eingestellten Koordinatensystem zu kartesischen Zielkoordinaten. Dabei werden jeder kartesischen Koordinate eigene kinematische Bewegungsgrößen zugeordnet.

Gleichzeitig können Sie die Hilfsachsen achsweise zu Zielachswerten verfahren.

Die folgende Abbildung zeigt 2 mögliche Bahnverläufe des Werkzeugmittelpunkts (TCP) bei Bewegung einer SCARA-Kinematik zwischen den Punkten A und B:



9308533643

Bei dem oberen Bahnverlauf sind die kartesischen Profile nicht synchronisiert.

Bei dem unteren, geraden Bahnverlauf sind die kartesischen Profile (Bewegung entlang X, Y) synchronisiert (*AxisGroupKin.Inst[..].In.Target.Synchronization[0]* = TRUE, alle Freiheitsgrade gleiche Ruckzeit *AxisGroupKin.Inst[n].In.Target.Cart.Jerk[i]*). Bei eingestellter Synchronisation ergibt sich in diesem Fall bei der Rast-in-Rast-Bewegung eine Gerade.

Ohne Synchronisation verfährt jedes Teilprofil (X, Y) zu den kartesischen Zielkoordinaten. Bei der nicht synchronisierten Bewegung von links unten nach rechts oben wird die Y-Koordinate früher erreicht, so dass anschließend nur noch entlang X zur Zielkoordinate verfahren wird.

20211732 / DE – 07/2014

9.3.6 TARGET-Überschleifen

Prinzip

Bei den TARGET-Betriebsarten erfolgt das Überschleifen durch Prüfung der gewünschten Überschleifdistanz zum Eckpunkt im Anwenderprogramm und die entsprechende Aktualisierung der Zielvorgabe bei Eintritt in den Überschleifbereich. Das heißt, sobald sich das Werkzeug in hinreichender Nähe des Zielpunkts befindet, wird ein neuer Zielpunkt vorgegeben, zu dem die Kinematiksteuerung automatisch hingeführt wird.

Bibliothek und Eingangssignale

Sie können zur Prüfung des Abstands zum aktuellen Zielpunkt die Funktion "MC_KinInProximity" aus der Bibliothek "MPLCKinematics" (→ 🗎 116) nutzen und als Transitionsbedingung in einer Schrittkette verwenden. In den Schritten werden jeweils die neuen Zielpositionen und Zielorientierungen in die Eingangsvariable *AxisGroupKin.Inst[n].In.Target.Position* geschrieben.

Das Eingangssignal "Threshold" der Funktion "MC_KinInProximity" stellt dabei die Entfernung vom Zielpunkt dar, oberhalb derer die Funktion den Wert "FALSE" zurückgibt. Bei genau dem Abstand "Threshold" oder einem kleineren Abstand wird der Wert "TRUE" zurückgegeben.

Im Eingangssignal "Dimension" spezifizieren Sie, welche kartesischen Dimensionen bei der Bestimmung der Entfernung einbezogen werden.

Funktionsaufruf für Translation und Rotation

Sollen die translatorische und die rotatorische Entfernung geprüft werden, genügt nur für den Fall, dass "Threshold = 0" ist, ein einzelner Funktionsaufruf. Ansonsten müssen für die Translation und die Rotation zwei voneinander unabhängige Aufrufe erfolgen. Das liegt daran, dass es nicht möglich ist die translatorische Entfernung (z. B. 10 mm) und die rotatorische Entfernung (z. B. 15°) in einem einzelnen Wert für den Eingang "Threshold" zu kombinieren.

Die folgenden Beispiele zeigen Ihnen Funktionsaufrufe für Translation und Rotation:

- **Translation**

  Check = 2#0000_0xxx (z. B. 2#0000_0111)

  Die X-, Y- und Z-Koordinaten werden in die Entfernung einbezogen (bei aktuell eingestelltem kartesischem Koordinatensystem).

- **Rotation**

  Check = 2#00xx_x000 (z. B. 2#0000_1000)

  Nur die A-Koordinate, also die Drehung um die Z-Achse, wird in die Bestimmung der (rotatorischen) Entfernung einbezogen (bei aktuell eingestelltem kartesischem Koordinatensystem).

20211732 / DE – 07/2014

9.4 ContinuousPath-Betriebsarten

9.4.1 Prinzip

Im Gegensatz zur TARGET-Interpolation wird bei der ContinuousPath-Interpolation eine Abarbeitungsliste, die sogenannte Queue, mit Bahnsegmenten befüllt. Die in die Queue eingetragenen Bahnsegmente werden entsprechend ihrer Parametrierung und der aktuell eingestellten prozentualen Abarbeitungsgeschwindigkeit bei aktivierter Vorschubfreigabe automatisch abgefahren.

Die ContinuousPath-Segmente werden im Eingangssignal *AxisGroupKin.Inst[..].In.Cp.Segment* eingetragen. Als Signal an die Kinematiksteuerung, ein ContinuousPath-Segment in die Queue zu übernehmen, wird eine sich ändernde ID (*AxisGroupKin.Inst[..].In.Cp.Segment.ID*) verwendet.

In einer Anwendung können somit jedem CP-Segment eindeutige IDs zugeordnet werden oder die ID wird einfach immer nur inkrementiert. Dazu würde auch ein einfacher Wechsel der ID zwischen "0" und "1" genügen. Davon wird jedoch abgeraten, da dann nur noch schwer von dem Ausgangssignal *AxisGroupKin.Inst[..].Out.Cp.Queue.ActSegID* auf das aktuell abgearbeitete CP-Segment geschlossen werden kann.

Ausgangssignale (beispielhaft) der Strukturvariablen *AxisGroupKin.Inst[..].Out.Cp*:

Ausgangssignale	Bedeutung
Queue.Size	Die aktuelle Anzahl der CP-Segmente in der Queue, die noch nicht abgearbeitet sind.
Path.Distance	Die bis zum Zielpunkt des zuletzt eingetragenen Segments noch zurückzulegende Strecke.
...	...

Ist das Eingangssignal *AxisGroupKin.Inst[..].In.General.Kin.FeedEnable* gesetzt, beginnt die Steuerung bereits beim Eintragen des ersten CP-Segments mit dessen Abarbeitung. Das heißt, dass bereits im ersten Zyklus die erforderlichen Sollwerte berechnet und an die Umrichter versendet werden.

In jedem Folgezyklus lässt sich ein weiteres CP-Segment in die Queue eintragen. Ebenso können bis zum letzten Zyklus der Abarbeitung einer Bahn weitere CP-Segmente eingetragen werden, so dass die Bewegung direkt wieder beschleunigt wird, um das angehängte CP-Segment ohne Unterbrechung abzufahren.

Die CP-Queue ist als Ringpuffer implementiert, so dass theoretisch endlos CP-Segmente ohne anzuhalten abgefahren werden können. Die einzige Beschränkung besteht in der Gesamtpfadlänge ab einem Zeitpunkt, zu dem die Queue komplett abgearbeitet und somit leer war.

Die Gesamtpfadlänge ist auf den Wert begrenzt, der durch eine 64 Bit Floating-Point Arithmetik maximal darstellbar ist. Das entspricht einem Wert von > 1000.000.000.000.000.000.000 Anwendereinheiten.

Als Standardwert hat die CP-Queue eine Größe von 256 Elementen (MOVI-PLC® power: 2048 Elemente), die während der Abarbeitung eingetragen werden können. Dabei ist jedoch zu beachten, dass beim Überschleifen (z. B. LIN -> LIN oder CIRC -> LIN) automatisch ein Überschleifbogen und das restliche Geradenstück bis zum Zielpunkt eingetragen werden. Hier werden also pro einzutragendem LINEAR-Segment zwei Queue-Elemente belegt. Werden nur LINEAR-Segmente eingetragen, können somit mehr als 100 Segmente auf einmal in der Queue für die Abarbeitung bereitgehalten werden.Die kinematischen Bewegungsgrößen für jedes einzelne CP-Seg-

20211732 / DE – 07/2014

ment werden individuell eingespeist (*Eingangssignal AxisGroupKin.Inst[..].In.Cp.Segment*). Während der Abarbeitung der Queue kann die Bewegung zusätzlich segmentübergreifend verlangsamt oder beschleunigt werden (Eingangssignal *AxisGroupKin.Inst[n..].In.Cp.Path.VelocityPercentage*).

9.4.2 KIN_LIN_XY / _YZ / _ZX

Die Betriebsarten **KIN_LIN_XY / _YZ / _ZX** ermöglichen die geometrisch exakt beschriebene Linearinterpolation des Werkzeugmittelpunkts (TCP) in den Hauptebenen des aktuell eingestellten Koordinatensystems.

Die Zielkoordinaten geben Sie in der Eingangsvariablen *AxisGroupKin.Inst[..].In.Cp.Segment.Position* an. Dabei wird die in der jeweils ausgewählten Betriebsart nicht verwendete Koordinate (z. B. Z-Koordinate bei **KIN_LIN_XY**) nicht aus dem Eingangssignal eingelesen, sondern aus dem zuletzt programmierten / erreichten Wert übernommen.

9.4.3 KIN_LIN_3D

Die Betriebsart **KIN_LIN_3D** ermöglicht die geometrisch exakt beschriebene Linearinterpolation des Werkzeugmittelpunkts (TCP) im 3-dimensionalen Raum im aktuell eingestellten Koordinatensystem.

Die Zielkoordinaten geben Sie in der Eingangsvariablen *AxisGroupKin.Inst[..].In.Cp.Segment.Position* an.

9.4.4 KIN_CIRC_XY / _YZ / _ZX

Die Betriebsarten **KIN_CIRC_XY / _YZ / _ZX** ermöglichen die Kreisinterpolation des Werkzeugmittelpunkts (TCP) in den Hauptebenen des aktuell eingestellten Koordinatensystems. Die Parametrierung eines Kreissegments erfolgt abhängig vom ausgewählten *AxisGroupKin.Inst[..].In.Cp.Segment.Circ.Mode* (siehe Kapitel "In.Cp.Segment" (→ 🗎 99)) z. B. mithilfe des Kreismittelpunkts und des zu überstreichenden Kreiswinkels.

Soll ein Kreisbogen in einer geneigten Ebene interpoliert werden, kann z. B. mithilfe der TARGET-Interpolation in ein Werkstückkoordinatensystem "KIN_PCSn" gewechselt und darin ein Kreissegment "KIN_CIRC_XY" programmiert werden. Die gewünschte Neigung können Sie in der Transformation *AxisGroupKin.Inst[..].In.Transform.WCS_PCSn* einstellen (siehe Kapitel "In.Transform" (→ 🗎 93)).

20211732 / DE – 07/2014

9.4.5 ContinuousPath-Überschleifen

Im Gegensatz zu TARGET-Betriebsarten wird das Überschleifen bei ContinousPath nicht durch Abstandsprüfung und entsprechende schrittweise Vorgabe neuer Zielpositionen angestoßen.

Stattdessen erfolgt die Programmierung des Überschleifens bei ContinuousPath-Betriebsarten durch Parametrierung der Strukturvariablen *AxisGroupKin.Inst[..].In.Cp.Segment.Blending* des Segments, zu dem übergeschliffen werden soll. Die Kinematiksteuerung leitet automatisch die Überschleifbewegung zum jeweils folgenden Segment ein.

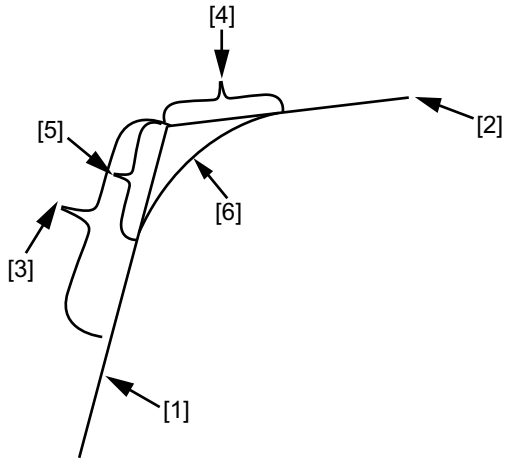
Eingangssignale der Strukturvariablen *AxisGroupKin.Inst[..].In.Cp.Segment.Blending*:

Eingangssignal	Funktion
Blending.Distance	Abstand vom letzten Zielpunkt , ab dem zum neuen Zielpunkt übergeschliffen werden soll, also der Radius der kugelförmigen Überschleifzone.
LimitationPercentage	Prozentsatz, um die programmierte *Blending Distance* automatisch zu reduzieren. Der eingegebene Prozentsatz bezieht sich auf den Abstand zwischen dem letzten Zielpunkt (der übergeschliffen wird) und dem Zielpunkt des Segments, das mit den Überschleifparametern im selben Zyklus übergeben wird.
Blending.Velocity	Geschwindigkeitsprofil und prozentuale Skalierung, mit dem der Überschleifbogen durchfahren werden soll (siehe Kapitel "In.Cp.Segment", Eingangssignal Blending (→ 🗎 100)).

20211732 / DE – 07/2014

Beispiel: Überschleifen LIN -> LIN

Die folgende Darstellung zeigt das Überschleifen von einem LINEAR-Segment auf ein weiteres LINEAR-Segment:



9735212939

[1] Letztes durch die Kinematiksteuerung übernommenes LINEAR-Segment

[2] Zielpunkt *Position* des neuen LINEAR-Segments (in aktuell gültiger Variablenstruktur AxisGroupKin.Inst[..].In.Cp.Segment)

[3] Parametrierte *Blending.Distance* (in aktuell gültiger Variablenstruktur AxisGroupKin.Inst[..].In.Cp.Segment)

[4] *Blending.LimitationPercentage* der Länge des neuen Geradenstücks (in aktuell gültiger Variablenstruktur *AxisGroupKin.Inst[..].In.Cp.Segment*)

[5] Resultierende Überschleifdistanz. Diese entspricht der parametrierten *Blending.Distance* [3]. Jedoch erstens begrenzt durch die zum Zeitpunkt der Übernahme des LINEAR-Segments verbleibende Reststrecke des letzten übernommenen LINEAR-Segments [1]. Und zweitens begrenzt durch die Blending.*LimitationPercentage* [4] der Länge des neuen Geradenstücks [2].

[6] Resultierender symmetrischer Überschleifbogen.

20211732 / DE – 07/2014

9.5 Vergleich von ContinuousPath- und TARGET- Überschleifen

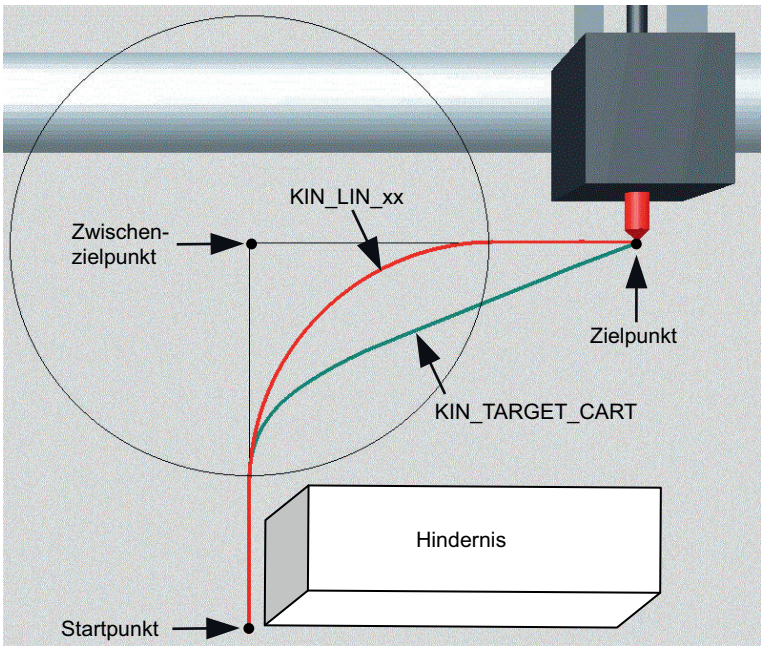
Bei ContinuousPath-Interpolation wird beim Überschleifen von einem Segment auf ein weiteres Segment ein symmetrischer Überschleifbogen eingefügt. D. h., dass der Überschleifbogen beim Eintritt in die kreisförmige Überschleifzone beginnt und beim Verlassen endet. Somit wird das programmierte Segment zum Zielpunkt nach dem Überschleifen erreicht, wie wenn nicht übergeschliffen würde.

Im Gegensatz hierzu wird bei TARGET-Interpolation beim Eintritt in die Überschleifzone, (wenn der vorgegebenen Abstands zum Zwischenzielpunkt unterschritten wird), ein neuer Zielpunkt vorgegeben. Ab diesem Zeitpunkt wird die Bewegung von der aktuellen Position und dem aktuellen Bewegungszustand mit den eingestellten Bewegungsparametern zum Zielpunkt hingeführt. Die Bahn zum Zielpunkt hängt von den Bewegungsparametern sowie der Einstellung der Synchronisation ab.

Es wäre bei TARGET-Interpolation ein Zufall, wenn die Bahn direkt vor Erreichen des Zielpunkts auf der Geraden verlaufen würde, auf welcher der Zielpunkt bei der Linear-Interpolation erreicht wird.

Stattdessen wird bei TARGET-Interpolation im Vergleich zur ContinuousPath-Interpolation abhängig von den Bewegungsparametern und der Einstellung der Synchronisation "stärker" oder "schwächer" sowie mit einer anderen Überschleifcharakteristik zum Zielpunkt abgebogen.

Die folgende Darstellung zeigt beispielhaft die Überschleifcharaktersitik für TARGET-Überschleifen im Vergleich zu ContinuousPath-Überschleifen:



9308546187

Die Zielgerade der TARGET-Interpolation (hier mit **KIN_TARGET_CAR** realisiert) liegt **unterhalb** der Zielgeraden der ContinuousPath-Interpolation (hier mit **KIN_LIN_xx** realisiert).

Je nach Einstellung ist auch der umgekehrte Fall möglich, sodass die Zielgerade der TARGET-Interpolation **oberhalb** der Zielgeraden der ContinuousPath-Interpolation verläuft.

Fazit

TARGET-Interpolation eignet sich sehr gut für die Umsetzung harmonischer Bahnen mit kurzen Zykluszeiten, wie sie i. d. R bei Handlingsvorgängen auftreten.

20211732 / DE – 07/2014

⚠ WARNUNG

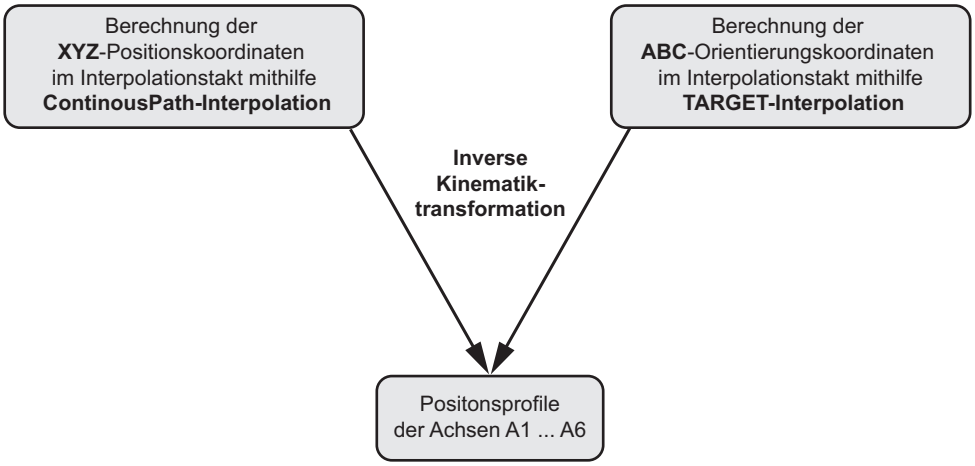
Durch die Änderung von Bewegungsparametern oder durch Stoppen und Wiederanfahren ergeben sich verschiedene Profile, die zu einer Bahn überlagert werden.

- Setzen Sie die TARGET-Interpolation nur dann ein, wenn sich die resultierende Bahn nicht im Bereich von Hindernissen befindet. Das ist bei der überwiegenden Anzahl von Handling-, Palettier-, Depalettier- und Tracking-Anwendungen der Fall, so dass die TARGET-Interpolation hier sehr gut einsetzbar ist.

9.6 Kombination von ContinuousPath- und TARGET-Betriebsarten

Sie können translatorische Bewegungen des Werkzeugmittelpunkts (TCP) mithilfe einer ContinuousPath-Interpolation und Orientierungsänderungen des Werkzeugkoordinatensystems (TCS) mithilfe einer TARGET-Interpolation (z. B. mit **KIN_TARGET_CART**) parallel ausführen.

Die folgende Darstellung zeigt, wie die unterschiedlich interpolierten Koordinaten mithilfe der Inversen Kinematiktransformation zu einem gemeinsamen Positionsprofil zusammengeführt werden:



9290278795

Diese kombinierte Betriebsart wird bei Wechsel in eine ContinuousPath-Betriebsart in *AxisGroupKin.Inst[..].In.Cp.Settings.PlusTargetCartABC* eingestellt (siehe Kapitel "In.Cp.Settings" (→ 🗎 99)).

Das Ausgangssignal *AxisGroupKin.Inst[..].Out.Cp.Settings.PlusTargetCartABC* zeigt an, ob die kombinierte Betriebsart aktiv ist (siehe Kapitel "Out.Cp" (→ 🗎 112)).

Die Kombination ist vorteilhaft in Anwendungen, in denen z. B. bei einem aus zahlreichen Segmenten bestehenden CP-Pfad nur jeweils die Orientierung im ersten und im letzten Segment von Interesse sind und auf die Angabe der Orientierungen in den Zwischenpunkten verzichtet werden soll. In dem Fall wird im Anwenderprogramm nach Durchfahren des ersten CP-Segments die Zielorientierung in die Eingangsvariable *AxisGroupKin.Inst[..].In.Target.Position* geschrieben. Das letzte CP-Segment darf erst in die Queue eingetragen werden, wenn im Anwenderprogramm erkannt wurde, dass die Zielorientierung erreicht ist (siehe Kapitel "Funktion MC_KinInProximity" (→ 🗎 116)).

20211732 / DE – 07/2014

Soll am Ende jedes CP-Segments eine bestimmte Zielorientierung erreicht werden, darf das jeweils folgende CP-Segment erst dann in die CP-Queue eingetragen und die nächste Zielorientierung erst dann in die Eingangsvariable *AxisGroupKin.Inst[..].In.Target.Position* geschrieben werden, wenn die aktuelle Zielorientierung erreicht ist.

Werden die folgende Position und Orientierung bereits etwas vor Erreichen der aktuellen Zielorientierung übergeben, ist ein Überschleifen der Orientierung realisiert.

Hierzu können Sie die Funktion "MC_KinInProximity" nutzen:

- Eingang Threshold = Orientierungsüberschleifdistanz
- Eingang Dimension = 2#0011_1000 (z. B. für Prüfung von ABC)

Das Überschleifen der translatorischen Bahn des Werkzeugmittelpunkts (TCP) wird auch in diesem Fall in *AxisGroupKin.Inst[..].In.Cp.Segment.Blending* angegeben.

20211732 / DE – 07/2014

10 MOVI-PLC®-Programm

10.1 Taskkonfiguration

Eine geeignete Taskkonfiguration wird beim Erstellen eines neuen PLC-Projekts automatisch angelegt, siehe Kapitel „Schritt 2: Erstellung eines MOVI-PLC®-Projekts (→ 🗎 27)“und „Schritt 3: Integration des Technologiemoduls (→ 🗎 28)“. Für das Technologie-Modul Kinematics sind die im Folgenden aufgelisteten Taskeinträge erforderlich. Wenn das PLC-Projekt nicht neu erstellt, sondern nur erweitert werden soll, können Sie diese Taskeinträge auch manuell hinzufügen.

- TaskMain

  → **PRG_TaskMain_Kinematics();**

  vor dem AxisHandler_Main_MultiMotion / _Light();

  Dieses Programm führt u. a. die folgenden Programme aus den Bibliotheken aus:

  – AxisGroupControl_Configuration_Kinematics(Enable := TRUE);

  – AxisGroupControl_Main_Kinematics(Enable := TRUE);

- TaskPriority

  → **PRG_TaskPriority_Kinematics();**

  vor dem AxisHandler_Priority_MultiMotion / _Light();

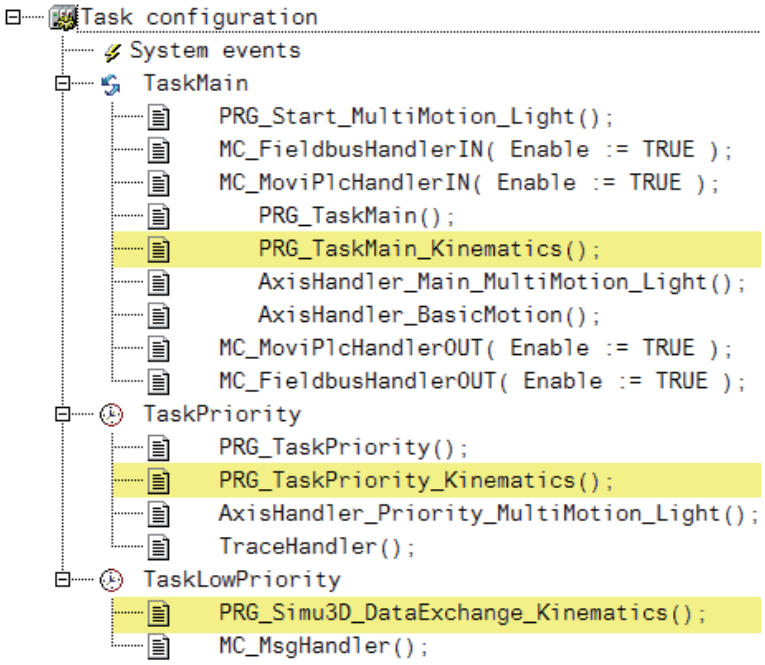
  Dieses Programm führt die folgenden Programme aus den Bibliotheken aus:

  – AxisGroupControl_Priority_Kinematics();

- TaskLowPriority

  → **PRG_Simu3D_DataExchange_Kinematics();**

  Dieses Programm ist in den Bibliotheken enthalten.



9310191755

20211732 / DE – 07/2014

10.2 Anwenderprogramm

Das Anwenderprogramm schreibt die für die gewünschten Abläufe und Bewegungsaufträge erforderlichen Werte, wie z. B. die Betriebsart sowie Bewegungsparameter, in die im Kapitel „Variablenschnittstelle AxisGroupKin.Inst[..] (→ 🗎 88)“ beschriebenen Eingangsvariablen *AxisGroupKin.Inst[..].In.*

⚠ WARNUNG

Damit die für einen Bewegungsauftrag verwendeten Variablen (z. B. Zielposition, Geschwindigkeit, Beschleunigung, …) konsistent durch die Kinematiksteuerung verwendet werden können, **MUSS** das Anwenderprogramm in der gleichen Task ausgeführt werden, in der auch das Programm *AxisGroupControl_Main_Kinematics* läuft. Gewöhnlich ist dies die TaskMain.

Wenn diese Voraussetzung verletzt ist, kann es zu inkonsistenten Bewegungsaufträgen und somit zu unvorhersehbaren Bewegungen der Kinematik kommen.

Ausnahme: Die folgenden Variablen müssen für Konsistenz in der TaskPriority beschrieben werden. Nur so ist es möglich, zyklisch im Interpolationstakt stetige Profile einzuspeisen, welche direkt für die Bewegungsführung verwendet werden.

- *AxisGroupKin.Inst[..].In.Transform* siehe Kapitel „In.Transform (→ 🗎 93)“
- *AxisGroupKin.Inst[..].In.MasterPosition* siehe Kapitel „In.MasterPosition (→ 🗎 94)“
- *AxisGroupKin.Inst[..].Config.Kin.Par*, wenn *AxisGroupKin.Inst[..].Config.CyclicTakeover.Kin_Par = TRUE*

⚠ WARNUNG

Selbst bei der durch das System gewährleisteten konsistenten Übernahme der Eingangsvariablen durch die Kinematiksteuerung gibt es Situationen, in denen im Anwenderprogramm (gewöhnlich TaskMain) vor dem Weiterschalten im Programmablauf explizit geprüft werden muss, ob die Kinematiksteuerung (gewöhnlich TaskPriority) seit dem letzten Ausführungszyklus des Anwenderprogramms ausgeführt wurde.

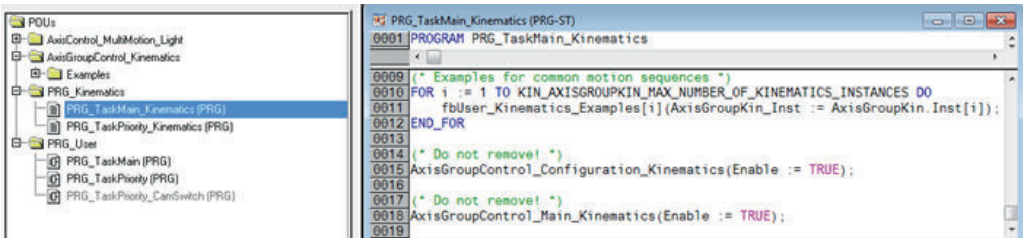
Diese Prüfung ist beispielsweise erforderlich, wenn bei ContinuousPath-Interpolation durch das Anwenderprogramm schrittweise Bahnsegmente in die Kinematiksteuerung eingetragen werden sollen. Die Segmentparameter werden zwar konsistent durch die Kinematiksteuerung übernommen. Wenn jedoch die TaskPriority zwischen 2 Zyklen des Anwenderprogramms nicht ausgeführt wurde und im Anwenderprogramm ohne Prüfung der Segmentübernahme in jedem Zyklus ein neues Bahnsegment parametriert würde, könnte es dazu führen, dass manche Bahnsegmente nicht durch die Kinematiksteuerung übernommen werden.

In dieser Situation muss im Anwenderprogramm geprüft werden, ob das zuletzt parametrierte Bahnsegment übernommen wurde, siehe Kapitel „ContinuousPath-Programmierung (→ 🗎 86)“ sowie Kapitel „Out.Cp (→ 🗎 112)“.

20211732 / DE – 07/2014

10.3 Beispielprogramme

Wenn Sie gemäß Kapitel „Schritt 2: Erstellung eines MOVI-PLC®-Projekts (→ 🗎 27)“ und „Schritt 3: Integration des Technologiemoduls (→ 🗎 28)“ ein neues PLC-Projekt erstellen, finden Sie Beispiele einfacher Anwenderprogramme im Ordner `AxisGroupControl_Kinematics/Examples`. Diese Beispielprogramme werden in den Funktionsbausteinen *fbUser_Kinematics_Examples* aufgerufen, dieses wiederum im Programm *PRG_TaskMain_Kinematics*.



9310150923

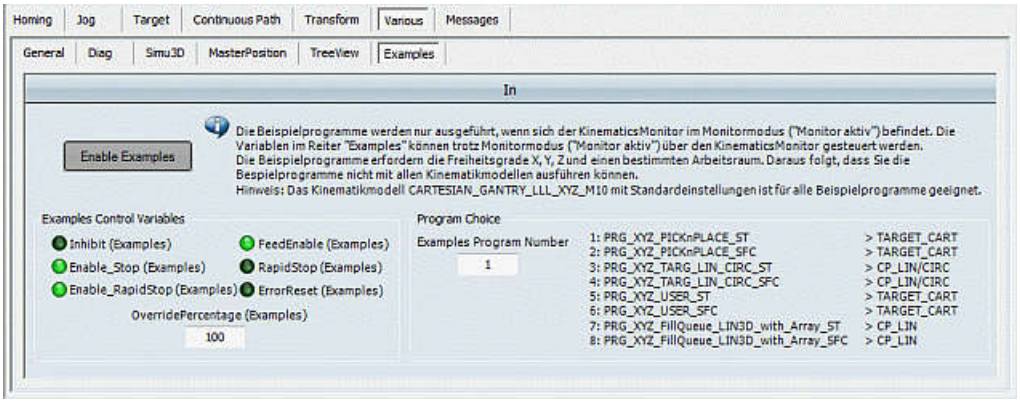
Alternativ können Sie Ihr Anwenderprogramm z. B. auch im *PRG_TaskMain* im Baustein-Ordner PRG_User oder an anderer Stelle in der TaskMain ausführen.

Sie können das Anwenderprogramm in jeder der verfügbaren IEC-Sprachen programmieren. Für die Umsetzung der erforderlichen Schrittketten eignen sich besonders z. B. die graphische Ablaufsprache (AS, engl. SFC) bzw. der strukturierte Text (ST).

Damit die im neu erstellten PLC-Projekt enthaltenen Beispielprogramme ausgeführt werden, müssen Sie die folgenden Variablen geeignet beschreiben.

gKinExamples_Control[..].EnableExamples	TRUE
gKinExamples_Control[..].ProgramNumber	1..8
gKinExamples_Control[..].Inverter.Inhibit_AllAxis	FALSE
gKinExamples_Control[..].Inverter.Enable_Stop_AllAxis	TRUE
gKinExamples_Control[..].Inverter.Enable_RapidStop_AllAxis	TRUE
gKinExamples_Control[..].Kin.FeedEnable	TRUE
gKinExamples_Control[..].Kin.RapidStop	FALSE
gKinExamples_Control[..].Kin.OverridePercentage	1..100

Sehr komfortabel können Sie die Beschaltung dieser Variablen in der Registerkarte “ Various / Examples” des Kinematikmonitors vornehmen:



12566156299

HINWEIS

Stellen Sie sicher, dass Ihr Applikationsprogramm **NICHT** startet, bevor *AxisGroupKin.ConfigDataAvailable* = TRUE. Ansonsten stehen die Konfigurationsdaten und Default-Belegungen für die Input-Variablen noch nicht zur Verfügung und werden eventuell überschrieben, wenn Ihr Applikationsprogramm diese Variablen (z. B. Override) zuvor beschreibt.

In den Beispielprogrammen erfolgt diese Prüfung im *fbUser_Kinematics_Examples* im Examples-Ordner.

10.3.1 TARGET-Programmierung

Bei Nutzung der TARGET-Interpolation werden neue Zielkoordinaten direkt nach Anlegen an die entsprechende Eingangsvariable angefahren (siehe Kapitel „In.Target (→ 🗎 95)“). Deshalb muss im Anwenderprogramm geprüft werden, ob ein anzufahrender Punkt bzw. dessen Nähe erreicht wurde, bevor die Zielkoordinaten des nächsten Punkts zugewiesen werden. Für diese Prüfung bietet sich die Funktion *MC_KinInProximity* an, siehe Kapitel „Funktion MC_KinInProximity (→ 🗎 116)“.

20211732 / DE – 07/2014

TARGET Beispielprogramm

```
…

CASE (MotionSequenceState) OF…

10: (* Heben *)

    (* Allgemeine Einstellungen, z.B. Betriebsart, Freigaben *)

    AxisGroupKin.Inst[1].In.General.Mode        := KIN_TARGET_CART;    (* Betriebsart *)

    AxisGroupKin.Inst[1].In.General.Inverter.Enable_RapidStop[0]  := …;   (* Freigabe Schnellstopp *)
    AxisGroupKin.Inst[1].In.General.Inverter.Enable_Stop[0]       := …;   (* Freigabe Stopp *)
    AxisGroupKin.Inst[1].In.General.Kin.FeedEnable                := …;   (* Kinematische Vorschubfreigabe *)

    (* Target Parametrierung *)

    AxisGroupKin.Inst[1].In.Target.Cart.Vel./Acc./Dec./Jerk[ …]      := …;   (* Kinematische Bewegungsgrößen
                                                                               für alle Dimensionen *)
    AxisGroupKin.Inst[1].In.Target.Synchronization[…]                := …;   (* Synchronisation von z.B. X,Y = 1,2 *)

    AxisGroupKin.Inst[1].In.Target.Position      := …;   (* Zielposition 1 *)
    AxisGroupKin.Inst[1].In.Target.CoordSys      := …;   (* Koordinatensystem *)

    (* Weiterschalten im Programmablauf, sobald Bewegung weit genug fortgeschritten ist *)

    IF MC_KinInProximity(…) THEN                  (* Abstandsprüfung, s. Kap.„ Funktion MC_KinInProximity“ *)
        MotionSequenceState := 20;
    END_IF

(*********************************************)

20: (* Transversal *)

    AxisGroupKin.Inst[1].In.Target.Position  := …;   (* Zielposition 2 *)

    IF MC_KinInProximity(…) THEN                  (* Abstandsprüfung *)
        MotionSequenceState := 30;
    END_IF

(*********************************************)

30: (* Senken *)

    AxisGroupKin.Inst[1].In.Target.Position  := …;   (* Zielposition 3 *)

    IF MC_KinInProximity(…) THEN                  (* Abstandsprüfung *)
        MotionSequenceState := 40;
    END_IF

(*********************************************)

40: (* … *) …



END_CASE

…
```

12511008523

(Siehe hierzu das Kapitel Beispiel: Pick-and-Place (→ 🗎 69)

10.3.2 ContinuousPath-Programmierung

Bei der CP-Interpolation werden die Bahnsegmente nacheinander in die Kinematiksteuerung eingespeist. Pro Zyklus der TaskMain kann 1 CP-Segment eingespeist werden. Bei erfolgreicher Prüfung der Parameter übernimmt die Kinematiksteuerung das jeweilige Segment und trägt es in eine Queue ein. Die Kinematiksteuerung führt dann automatisch die Bahninterpolation und das parametrierte Überschleifen zwischen den einzelnen, in die Queue eingetragenen Segmenten durch. Das Überschleifen wird also im Gegensatz zur TARGET-Interpolation nicht im Anwenderprogramm angestoßen.

20211732 / DE – 07/2014

ContinuousPath Beispielprogramm

```
…
CASE (ProgramState) OF…
10: (* Einspeisen des Heben-Segments *)
    (* Allgemeine Einstellungen, z.B. Betriebsart, Freigaben *)

    AxisGroupKin.Inst[1].In.General.Mode                    := KIN_LIN_3D;    (* Betriebsart *)

    AxisGroupKin.Inst[1].In.General.Inverter.Enable_RapidStop[0]  := …;   (* Freigabe Schnellstopp *)
    AxisGroupKin.Inst[1].In.General.Inverter.Enable_Stop[0]       := …;   (* Freigabe Stopp *)
    AxisGroupKin.Inst[1].In.General.Kin.FeedEnable                := …;   (* Kinematische Vorschubfreigabe *)

    (* ContinuousPath Parametrierung *)

    AxisGroupKin.Inst[1].In.Cp.Segment.Translation.Vel./Acc./Dec./Jerk     := …; (* Kinematik der Translation *)
    AxisGroupKin.Inst[1].In.Cp.Segment.Blending.Distance                   := …; (* Überschleif-Distanz *)
    AxisGroupKin.Inst[1].In.Cp.Segment.Blending.LimitationPercentage      := 50; (* Überschleif-Prozentsatz *)

    AxisGroupKin.Inst[1].In.Cp.Segment.Position    := …;    (* Zielposition 1 *)
    (* ID ändern, z.B. inkrementieren, als Signal für die Kinematiksteuerung, das Segment zu übernehmen *)
    AxisGroupKin.Inst[1].In.Cp.Segment.ID  := AxisGroupKin.Inst[1].Out.Cp.Queue.LastMappedSegID + 1;

    ProgramState:= 20;
(*********************************************)
20: (* Einspeisen des Transversal-Segments *)
    (* Weiterschalten in der Befüllung der ContinuousPath Queue
       Das oder die zuletzt in die Kinematiksteuerung übernommenen Segmente sind zu dem Zeitpunkt i. d. R.
       noch nicht oder nur zu einem Bruchteil abgefahren, d. h. der Roboter kann sich in Bewegung befinden. *)

    IF (AxisGroupKin.Inst[1].Out.Cp.Queue.LastMappedSegID = AxisGroupKin.Inst[1].In.Cp.Segment.ID) THEN
        (* Segment wurde übernommen → nächstes einspeisen, siehe auch Warnung in Kap. „Anwenderprogramm“ *)

        AxisGroupKin.Inst[1].In.Cp.Segment.Position    := …;    (* Zielposition 2 *)
        AxisGroupKin.Inst[1].In.Cp.Segment.ID  := AxisGroupKin.Inst[1].In.Cp.Segment.ID + 1;   (* ID ändern *)
        ProgramState:= 30;
    END_IF
(*********************************************)
30: (* Einspeisen des Senken-Segments *)

    IF (AxisGroupKin.Inst[1].Out.Cp.Queue.LastMappedSegID = AxisGroupKin.Inst[1].In.Cp.Segment.ID) THEN
        (* Segment wurde übernommen → nächstes einspeisen *)

        AxisGroupKin.Inst[1].In.Cp.Segment.Position    := …;    (* Zielposition 3 *)
        AxisGroupKin.Inst[1].In.Cp.Segment.ID  := AxisGroupKin.Inst[1].In.Cp.Segment.ID + 1;   (* ID ändern *)
        ProgramState:= 40;
    END_IF
(*********************************************)
40: (*  Warten auf Erreichen des Bahn-Endes
        Bis dahin muß AxisGroupKin.Inst[1].In.General.Mode ein CP-Mode sein. Es können also LIN- und CIRC-Segmente
        eingespeist werden, die dann durch die Kinematiksteuerung automatisch abgefahren werden. Sobald jedoch ein anderer
        Mode angelegt wird, z.B. KIN_STOP/JOG/TARGET, AM_DEFAULT, AM_HOMING, ist der Inhalt der CP-Queue gelöscht. *)

    IF (AxisGroupKin.Inst[1].Out.Cp.Queue.LastMappedSegID = AxisGroupKin.Inst[1].In.Cp.Segment.ID)
        (* Segment wurde übernommen *)
    AND (AxisGroupKin.Inst[1].Out.Cp.Path.EndReached) THEN
        (* Bahn komplett abgefahren *)

        ProgramState:= …; (* u. s. w. *)

    END_IF
END_CASE
```

12511013003

11 Variablenschnittstelle AxisGroupKin.Inst[..]

Die komplette Steuerung aller Kinematikinstanzen erfolgt über die globale Strukturvariable AxisGroupKin, die im Baustein GlobalVar_AxisGroupControl_Kinematics im Ordner AxisGroupControl_Kinematics deklariert ist.

Variable	Typ	Bedeutung		
HMI	ST_AxisGroupControl_HMIInterface (03_ApplicationModules \ MPLCAxisGroupControl.lib)	Diese Variable darf NICHT durch den Anwender beschrieben werden. Sie zeigt den Zustand des HMI-Steuerbetriebs im Kinematik-Monitor an. **Hinweis:** Wenn die Steuerung exklusiv aus dem Kinematik-Monitor erfolgen soll, darf das Applikationsprogramm NICHT ausgeführt werden bzw. die globale Variable AxisGroupKin NICHT beschreiben, wenn HMI.HMIControl = TRUE (siehe auch die mit dem Technologie-Modul " Kinematics" importierte Beispiele, Baustein PRG_User_Kinematics_Examples).		
		Detailinformation:		
		HMIControl	BOOL	HMI Steuerbetrieb ist im Monitor aktiviert
		WatchDogIn	BOOL	Spiegel vom Engineering-PC
		WatchDogOut	BOOL	Toggle Bit von der MOVI-PLC®
		WatchDogTimeout	BOOL	WatchDog in TimeOut
		WatchDogTimeOutTime	DWORD	TimeOut Zeit [ms]
NumberOfInstances	UINT	Anzahl der konfigurierten und somit aktivierten Kinematikinstanzen (siehe auch *Variable AxisGroupKin.Inst[..].Config.Enable*).		
ConfigDataAvailable	BOOL	Die Konfigurations-Daten und Default-Werte für Input-Variablen für alle Kinematikinstanzen und alle Achsen wurden von der Speicherkarte gelesen und den entsprechenden Variablen zugewiesen. **Hinweis:** Solange *ConfigDataAvailable* = FALSE, sollte das Applikationsprogramm NICHT ausgeführt werden (siehe Kapitel „Beispielprogramme (→ 🗎 83)").		
Inst	ARRAY [1..] OF ST_AxisGroupKin_Instance (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)	Konfigurations-, Input-, Output- und HMI-Daten der Kinematikinstanzen.		
		Für jede Instanz stehen die folgenden Variablen zur Verfügung:		
		Config	ST_AxisGroupKin_Config	Automatisch durch das System beschrieben
		In	ST_AxisGroupKin_In	Kinematiksteuervariablen (siehe unten)
		Out	ST_AxisGroupKin_Out	Kinematikanzeigevariablen (siehe unten)
		HMI	ST_AxisGroupKin_HMIGeneral	HMI-Variablen, vom Monitor verwendet

20211732 / DE – 07/2014

HINWEIS

Die Kinematiksteuerung erfolgt ausschließlich mittels der globalen Variablenschnittstelle „AxisGroupKin“.

Der Anwender bzw. das Applikationsprogramm darf die Variablenschnittstelle „AxisInterface“ der zur Kinematik gehörenden MultiMotion-Achsen NICHT beschreiben.

20211732 / DE – 07/2014

11.1 In.General

Typ: MC_KIN_IN_GENERAL (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung		
Inverter	MC_KIN_IN_GENERAL_INVERTER (04_InterpolatedMotion\MPLCKinematics.lib)	Inhibit	ARRAY[0..8] OF BOOL	Reglersperre
		Enable_RapidStop	ARRAY[0..8] OF BOOL	MultiMotion Freigabe/Schnellstopp
		Enable_Stop	ARRAY[0..8] OF BOOL	MultiMotion Freigabe/Stopp
		Die Bits 1..8 beziehen sich auf die Achsen 1..8. Bit 0 bezieht sich auf alle konfigurierten Achsen dieser Kinematik-Instanz. **Hinweis:** Wenn Enable_RapidStop oder Enable_Stop bei mindestens einer konfigurierten Achse einer Kinematikinstanz auf FALSE gesetzt wird, werden alle Achsen dieser Instanz nicht mehr kinematisch geführt, sondern durch MultiMotion mittels eines Stopp-Profils abgebremst. Somit wird i.allg. NICHT bahntreu abgebremst. **Ausnahme:** In der Betriebsart KIN_JOG_AXIS können auch einzelne Kinematikachsen freigegeben und getippt werden, unabhängig vom Status „referenziert“.		
Kin	MC_KIN_IN_GENERAL_KIN (04_InterpolatedMotion\MPLCKinematics.lib)	FeedEnable	BOOL	
		FALSE	Kinematisches Bremsen der Bewegung bis zum Stillstand im aktuellen Koordinatensystem mit den Rampen des aktuellen Bewegungsauftrags	
		TRUE	Ausführung des aktuellen Bewegungsauftrags	
		RapidStop	BOOL	
		TRUE	Kinematisches Bremsen der Bewegung bis zum Stillstand im aktuellen Koordinatensystem mit den konfigurierten Schnellstopp-Rampen	
		FALSE	Ausführung des aktuellen Bewegungsauftrags	
		Achtung: In einer kartesischen Betriebsart (z. B. KIN_TARGET_CART, KIN_LIN_3D) z. B. in einem bewegten Koordinatensystem PCS wird die Bewegung des Tools beim Rücksetzen von FeedEnable bzw. beim Setzen relativ zum aktuellen, bewegten PCS abgebremst. Die Achsen bewegen sich dann i. a. noch. Wenn in dem Fall die Achsen zum Stillstand kommen sollen, muss applikativ in Interpolation der Achsen (z. B. KIN_JOG/TARGET_AXIS) umgeschaltet werden. Siehe auch Hinweis im Kapitel „Out.General (→ 🗎 107)“, Ausgangsvariable „FBError“.		
		OverridePercentage	UINT	1 .. konfigurierter Maximalwert in [%]
Ignore	MC_KIN_IN_GENERAL_IGNORE (04_InterpolatedMotion\MPLCKinematics.lib)	Deaktivierung von Überwachungen		
		Inverter	ARRAY[0..8] OF BOOL	
		Wenn TRUE, wird die Existenz sowie der Zustand der entsprechenden Achsen ignoriert.		

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
		Konfigurierte, jedoch nicht vorhandene oder nicht betriebsbereite, ignorierte Achsen werden trotzdem virtuell interpoliert, d. h. in den Bahnberechnungen mit einbezogen. Im Gegensatz zur Simulation der Achsen wird der tatsächliche Status der Achsen im OUT-Bereich angezeigt. Ignorierte Achsen werden durch die Kinematiksteuerung in AM_DEFAULT geschaltet und Enable_Stop sowie Enable_RapidStop dieser Achsen werden auf FALSE gesetzt. Die Bits 1..8 beziehen sich auf die Achsen 1..8. Bit 0 bezieht sich auf alle konfigurierten Achsen dieser Kinematikinstanz.		
		SWLS.Axis	BOOL	Wenn TRUE, wird KEIN Fehler generiert bei Verlassen der konfigurierten Software-Endschalter mindestens einer Achse (A1 .. A8), mindestens einer kartesischen Koordinate (XYZABC) oder mindestens einer Modell-abhängigen Kinematiklimitierung (z. B. Winkel eines Gelenks in einer Parallelkinematik, dem kein Antrieb zugeordnet ist)
		SWLS.Cart	BOOL	
		SWLS.Kin	BOOL	
		MotorSpeed	BOOL	Wenn TRUE, wird kein Fehler generiert bei Überschreitung der konfigurierten, maximal zulässigen Motordrehzahl
		KinHandicap	BOOL	Wenn TRUE, werden kartesische Freiheitsgrade in der inversen kinematischen Berechnung ignoriert, welche das Kinematik-Modell nicht unterstützt. Zum Beispiel werden die Orientierungs-Koordinaten ABC bei einem Modell ignoriert, das den Flansch FCS relativ zum KCS Bezugskoordinatensystem nicht verdrehen kann.

20211732 / DE – 07/2014

Variable	Typ	Bedeutung
Mode	MC_KIN_MODE	Betriebsart für alle Achsen der Kinematikinstanz. KIN_AM_DEFAULT KIN_AM_HOMING, KIN_STOP KIN_JOG_AXIS, KIN_JOG_CART, KIN_TARGET_AXIS, KIN_TARGET_CART, KIN_LIN_XY, KIN_LIN_YZ, KIN_LIN_ZX, KIN_LIN_3D, KIN_CIRC_XY, KIN_CIRC_YZ, KIN_CIRC_ZX **Hinweis:** Alternativ kann an diese Eingangsvariable statt einer konkreten ContinuousPath-Betriebsart (z. B. KIN_LIN_XY, KIN_CIRC_ZX, …) auch die übergreifende Betriebsart KIN_CONTINUOUS_PATH angelegt werden. In dem Fall muss die konkrete ContinuousPath-Betriebsart in der ContinuousPath Segment-Struktur eingegeben werden, siehe Kapitel „In.Cp.Segment (→ 🗎 100)“. Der Vorteil liegt darin, dass die CP-Segmente des Datentyps MC_KIN_IN_CP_SEGMENT mit der kompletten Segmentbeschreibung inklusive dem konkreten CP-Mode im Applikationsprogramm z. B. aus einem Array von CP-Segmenten einfach komplett auf den Eingang *In.Cp.Segment* der Kinematiksteuerung kopiert werden können.
Reset	BOOL	Bei steigender Flanke werden Fehler zurückgesetzt. **Hinweis:** Zusätzlich werden bei steigender Flanke auch die Variablen Axis.PastAbsoluteReachedMotorSpeed SWLS.Past_Axis_Cart_Kin SWLS.PastMotorSpeedWarning SWLS.PastMotorSpeed in der Ausgangsvariablen *Out.General* zurückgesetzt, siehe Kapitel „Out.General (→ 🗎 107)“. Diese Variablen werden auch beim Wechsel in eine Kinematik-Betriebsart (z. B. nachdem eine Achse nicht betriebsbereit war) sowie im Mode KIN_AM_DEFAULT zurückgesetzt.

11.2 In.Homing

Typ: ST_AxisGroupKin_In_Homing (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)

Variable	Typ	Bedeutung
Start	ARRAY[0..8] OF BOOL	Bei steigender Flanke wird das konfigurierte Referenzieren der entsprechenden Achse(n) gestartet. Das Signal muss während des Referenzierens gesetzt bleiben, bis das entsprechende Bit in der Ausgangsvariablen Out.Homing.Done gleich TRUE wird, siehe Kap. „Out.Homing (→ 🗎 112)“. Bei verfrühtem Rücksetzen des Eingangssignals wird das Referenzieren dieser Achse abgebrochen. Die Bits 1..8 beziehen sich auf die Achsen 1..8. Bit 0 bezieht sich auf alle konfigurierten Achsen dieser Kinematikinstanz.

20211732 / DE – 07/2014

11.3 In.Transform

Typ: MC_KIN_IN_TRANSFORM (04_InterpolatedMotion\MPLCKinematics.lib)

Die Eingangs-Variablen in dieser Struktur ermöglichen die Einstellung der Transformationen zwischen den Koordinatensystemen der Kinematiksteuerung. Wenn die während der Roboter-Bewegung maßgeblichen Transformationen verändert werden sollen (z. B. Tracking auf bewegtes Werkstück-Koordinatensystem PCS1), müssen die Zuweisungen mit stetigem Signalverlauf zyklisch in der TaskPriority erfolgen.

Die Array-Elemente 1..3 entsprechen den translatorischen Versätzen entlang X,Y,Z des Bezugskoordinatensystems, also WCS bei den Transformationen WCS_KCS/ PCS1/2 bzw. FCS bei der Tool-Transformation. Die Array-Elemente 4..6 spezifizieren die Drehung um die Z-Achse des Bezugskoordinatensystems, die anschließende Drehung um die neue (also um Z verdrehte) Y-Achse und die folgende Drehung um die wiederum neue (also um Z und dann um Y verdrehte) X-Achse.

Eine Übersicht der Koordinatensysteme sowie weiterführende Information finden Sie im Kapitel „Wechsel des Koordinatensystems (→ 🗎 127)“.

Variable	Typ	Bedeutung
WCS_KCS	ARRAY[1..6] OF LREAL	Transformation vom World Coordinate System zum Kinematics Coordinate System
WCS_PCS1	ARRAY[1..6] OF LREAL	Transformation vom World Coordinate System zum Piece Coordinate System 1
WCS_PCS2	ARRAY[1..6] OF LREAL	Transformation vom World Coordinate System t zum o Piece Coordinate System 2
Tool	ARRAY[1..6] OF LREAL	Transformation vom Flange Coordinate System zum Tool Coordinate System

Die Einheit der Komponenten 1..3 ist die konfigurierte [translatorische Anwendereinheit].

Die Einheit der Komponenten 4..6 ist die konfigurierte [rotatorische Anwendereinheit] = [rad] or [degr].

ACHTUNG

Unstetigkeiten der Signale in diesen Variablen haben folgende Konsequenzen zur Folge:

- Kollisionsgefahr durch unerwartet große Kurvenverläufe (z. B. beim Wechsel zwischen Koordinatensystemen oder Betriebsarten).
- Drehzahlüberschreitungen
- Brummende Antriebe
- Schleppfehler

- Bereits 5 Zyklen der TaskPriority vor der Umschaltung in ein neues Koordinatensystem müssen die dann relevanten Transformationen stetig angelegt werden.

20211732 / DE – 07/2014

11.4 In.MasterPosition

Es ist bei Target und ContinuousPath Interpolation möglich, die Profilgenerierung abzuschalten und statt dessen einem von außen eingespeisten Bewegungsprofil zu folgen (siehe auch Kap. „Kombination von Kurvenscheiben- und Kinematikfunktionalität (→ 🗎 136)“ und Kap. „CP-Interpolation als Slave eines externen Masterprofils (→ 🗎 136)“). Die Bewegungsprofile werden in der Task Priority an die hier beschriebenen Eingänge angelegt.

Typ: MC_KIN_IN_MASTERPOSITION (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
Target	ARRAY[1..8] OF LREAL	Optionale Master Target-Position für TARGET Betriebsarten Verwendet, wenn In.Target.UseMasterPosition.XYZABC_OR_A1ToA6 = TRUE, siehe Kap. „In.Target (→ 🗎 95)“ Position in [Anwendereinheit], Orientierung in [Anwendereinheit] = [rad] or [degr]
Cp	LREAL	Optionale Master-Position für CP Bahnen Verwendet, wenn In.Cp.Settings.UseMasterPosition = TRUE beim Wechsel (von einer anderen Betriebsart als CP) in eine CP-Betriebsart, siehe Kap. „In.Cp.Settings (→ 🗎 99)“ Entspricht der zurückgelegten Strecke entlang der CP-Bahn [Anwendereinheit] Kann mit beliebigem Wert beginnen Muss bis zum Erreichen des Pfadendes auf Stillstand abgebremst sein, da die Bewegung ansonsten schlagartig beendet wird. Kann stoßfrei auf den Wert 0.0 zurückgesetzt werden, wenn zuvor > 0 Wenn die Werte dekrementiert werden, darf der Wert 0 NICHT erreicht werden, da dies als Zurücksetzen interpretiert würde und deshalb trotz Anlegens von 0 der Wert des letzten Zyklus erhalten bliebe. Eine Rückwärts-Bewegung muss also VOR Erreichen der CP-MasterPosition 0 abgebremst sein. Ansonsten würde die Zuordnung einer Master-Position zu einer Raumposition bei mehrmaligem Vor- und Zurückbewegen wegdriften.

HINWEIS

Unstetigkeiten der Signale in diesen Variablen führen bei Verwendung zur Bewegungssteuerung zu:

- Brummende Antriebe
- Schleppfehler

20211732 / DE – 07/2014

11.5 In.Jog

Typ: MC_KIN_IN_JOG (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
Positive	ARRAY[1..8] OF BOOL	Positives Tippen der bis zu 8 Freiheitsgrade (A1 .. A8 bei KIN_JOG_AXIS, XYZABC + A7/8 bei KIN_JOG_CART)
Negative	ARRAY[1..8] OF BOOL	Negatives Tippen der bis zu 8 Freiheitsgrade (A1 .. A8 bei KIN_JOG_AXIS, XYZABC + A7/8 bei KIN_JOG_CART)
CoordSys	MC_KIN_COORDSYS	Koordinatensystem, in dem bei KIN_JOG_CART kartesisch getippt werden soll: KIN_KCS, KIN_TCS, KIN_WCS, KIN_PCS1, KIN_PCS2 Die Einstellung KIN_ACS führt bei KIN_JOG_CART zum Tippen in KIN_KCS, jedoch Anzeige der aktuellen Position in *Out.General.Position.ActCoordSys* in ACS. Bei KIN_JOG_AXIS werden immer die Achsen getippt. Die Anzeige der Position in *Out.General.Position.ActCoordSys* erfolgt im eingestellten Koordinatensystem. Bei Einstellung von KIN_TCS erfolgt die Anzeige der aktuellen Position unabhängig von der Tipp-Betriebsart in KCS.
VelocityPercentage	UINT	Prozentsatz der konfigurierten Tipp-Geschwindigkeiten 0 .. konfigurierter Maximalwert in [%], Defaultwert 100%

11.6 In.Target

Typ: MC_KIN_IN_TARGET (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung	
Position	ARRAY[1..8] OF LREAL;	Position:	[translatorische Anwendereinheit],
		Orientierung:	[rotatorische Anwendereinheit] = [rad] oder [degr]
CoordSys	MC_KIN_COORDSYS	KIN_KCS, KIN_ACS, KIN_TCS, KIN_WCS, KIN_PCS1, KIN_PCS2 Sie finden eine Übersicht der Koordinatensysteme im Kapitel „Wechsel des Koordinatensystems (→ 🗎 127)“	
Constellation	UINT	Konstellation, in welcher eine kartesische Target-Position in der Betriebsart KIN_TARGET_AXIS angefahren werden soll,	
		0	Aktuelle Konstellation
		1, 2	Explizit definierte Konstellation (z. B. bei SCARA: Ellbogen links/rechts)
		Die aktuelle Konstellation wird in der Ausgangsvariable *Out.General.Kin.Constellation* angezeigt, siehe Kapitel „Out.General (→ 🗎 107)“. Ein Wechsel der Konstellation ist nur in Achsen-Interpolation (KIN_JOG/TARGET_AXIS) oder z. B. durch manuelles Verschieben der Achsen bei abgeschalteter Maschine möglich.	

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
AxisPhase	ARRAY[1..8] OF DINT	Phasen der rotativen Achsen, in welchen eine kartesische Target-Position in der Betriebsart KIN_TARGET_AXIS angefahren werden soll. Unter einer rotatorischen Achse wird hierbei eine Achse verstanden, welche einen Drehwinkel zwischen zwei Körpern der kinematischen Kette verändert. Addiert man zu diesem Drehwinkel ganzzahlige Vielfache von 360°, führt dies zur gleichen kartesischen Stellung des Roboters. In der Variablen „AxisPhase“ können Sie explizit einstellen, in welches ganzzahlige Vielfachen von 360°, also in welche Achsen-Phase der Roboter verfahren soll.		
		0	Automatische Auswahl der am nächst gelegenen Phase, mit welcher die Target-Position angefahren werden kann, ohne die Software-Endschalter der Achsen zu verletzen.	
		andere Werte	Explizite Auswahl, Beispiel: -5 bedeutet, dass die entsprechende Achse in die Phase]-4 * 360°, -5 * 360°] verfahren soll.	
		Die Array-Elemente 1 ..8 beziehen sich auf die Achsen 1..8. Die Variable bzw. Array-Elemente werden NICHT verwendet: • Für Linearachsen (d. h. Achsen, die zu einer linearen Bewegung führen, unabhängig vom Antriebstyp) • In anderen Betriebsarten als KIN_TARGET_AXIS • Wenn *Config.General.AxisCartBijection* = TRUE (In dem Fall werden die Phasen der Achsen automatisch eindeutig ermittelt. AxisCartBijection können Sie in der MotionStudio Berechtigungsstufe „Advanced“ auf der Expertenseite der Konfiguration einstellen.)		
UseMasterPosition	MC_KIN_IN_TARGET_USEMASTERPOSITION	Wenn UseMasterPosition.XYZABC_OR_A1ToA6 = TRUE ist, werden für die Hauptachsen A1..A6 keine Verfahrprofile erzeugt. Statt dessen wird die Target-Position am Eingang In.MasterPosition.Target im Zyklus der TaskPriority angesteuert, siehe Kapitel „In.MasterPosition (→ 🗎 94)“. Die Target-Position ist abhängig vom eingestellten Koordinatensystem „In.Target.CoordSys“ in kartesischen Koordinaten oder in Achswerten ausgedrückt. Diese Funktionalität ermöglicht es z. B. online gerechnete Kurven, Stützstellentabellen oder auch Anwenderprofile mit einer Kinematik abzufahren. Sobald die Variable UseMasterPosition auf FALSE zurückgesetzt wird, setzt die Kinematiksteuerung ruckfrei auf die bis dahin von außen eingespeiste Master-Bewegung auf und führt die Bewegung mittels intern erzeugter Verfahrprofile fort. Die von außen eingespeisten Kurven können in jedem Koordinatensystem ausgeführt werden, z. B. auch relativ zu einem bewegten Werkstück in einer Tracking-Anwendung. Eine weitere interessante Möglichkeit besteht in einer speziellen, applikativen ABC-Orientierungsführung während ContinuousPath Interpolation. Siehe auch Kap. „Kombination von Kurvenscheiben- und Kinematikfunktionalität (→ 🗎 136)“.		
Axis	MC_KIN_IN_TARGET_AXIS	Kinematische Bewegungsgrößen für die Betriebsart KIN_TARGET_AXIS		
		Velocity	ARRAY[1..8] OF LREAL	[Anwendereinheit / sec], >= 0

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
		Acceleration	ARRAY[1..8] OF LREAL	[Anwendereinheit / (sec*sec)], > 0
		Deceleration	ARRAY[1..8] OF LREAL	[Anwendereinheit / (sec*sec)], >0, <= *Config.Axis.RapidDeceleration[1..8]*
		Jerk	ARRAY[1..8] OF LREAL	[ms], >= *Config.Axis.RapidJerk[1..8]*
		Hinweis: Die Ausgangsvariable *Out.Diag.MaxAllowedAxisVelocity* gibt für alle Achsen die entsprechend der konfigurierten Motordrehzahlen maximal zulässigen Achsen-Geschwindigkeiten an.		
Cart	MC_KIN_IN_TARGET_CART	Kinematische Bewegungsgrößen für die Betriebsart KIN_TARGET_CART		
		Velocity	ARRAY[1..6] OF LREAL	[Anwendereinheit / sec], >= 0
		Acceleration	ARRAY[1..6] OF LREAL	[Anwendereinheit / (sec*sec)], > 0
		Deceleration	ARRAY[1..6] OF LREAL	[Anwendereinheit / (sec*sec)], >0, <= *Config.Cart.RapidDeceleration[1..8]*
		Jerk	ARRAY[1..6] OF LREAL	[ms], >= *Config.Cart.RapidJerk[1..8]*
Synchronization	ARRAY[0..8] OF BOOL	Die Freiheitsgrade, für welche die entsprechenden Bits auf TRUE gesetzt sind, werden synchronisiert. In der Betriebsart KIN_TARGET_AXIS beziehen sich die Array-Elemente 1..6 auf die Achsen A1..A6. In der Betriebsart KIN_TARGET_CART beziehen sich die Array-Elemente 1..6 auf die kartesischen Freiheitsgrade XYZABC. Bit 0 = TRUE führt zur Synchronisation aller Freiheitsgrade. Die Synchronisation kann auch während der Bewegung ein- oder ausgeschaltet bzw. geändert werden. **Hinweise:** • In einer mittels KIN_TARGET_CART realisierten Pick'n'Place-Bewegung ist es i. a. sinnvoll, die vertikale Raumkoordinate, also z. B. Z, aus der Synchronisation auszunehmen und nur z. B. X und Y zu synchronisieren. • Damit sich bei einer synchronisierten Rast-in-Rast Bewegung mittels KIN_TARGET_CART eine Gerade ergeben kann, müssen alle Komponenten mit der gleichen Ruckzeit eingestellt sein (*In.Target.Cart.Jerk[i]*).		
ModuloMode	ARRAY[1..8] OF MC_KIN_MODULO_MODE	In Vorbereitung		

Variable	Typ	Bedeutung	
RapidDynamicsAtOvershooting	BOOL	TRUE	Bei aktuellem oder absehbarem Überschwingen über eine Zielkoordinate hinaus werden die konfigurierten Schnellstopp-Rampen zum Abbremsen verwendet.
		FALSE	Keine Aktion
ABCMapping	BOOL	TRUE	Im Zyklus des Wechsels in die Betriebsart KIN_TARGET_CART hinein oder des Wechsels des Koordinatensystems (KCS, WCS, PCS2, PCS2) innerhalb der Betriebsart KIN_TARGET_CART wird die AKTUELLE kartesische Orientierung ABC (Array-Elemente 4..6) in die Phase gemappt, in welcher der Orientierungswert am nähesten am Orientierungswert in der Variablen „In.Target.Position[4..6]“ in diesem Zyklus ist.
		FALSE	Keine Aktion
		Hinweis: Mittels dieser Funktion können Sie die absolute kartesische Orientierung einstellen, welche Sie in Ihrer Anwendung in der entsprechenden Situation und Stellung des Roboters erwarten. Ausgehend von dieser absoluten kartesischen Orientierung können Sie in Ihrem Ablaufprogramm klar definiert absolute kartesische Orientierungswerte anfahren. Kapitel „Umgang mit Mehrdeutigkeiten der ABC-Orientierungswerte (→ 🗎 138)“ geht detailliert auf die Hintergründe dieser Funktionalität ein.	

11.7 In.Cp

Typ: MC_KIN_IN_CP (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
Settings	MC_KIN_IN_CP_SETTINGS	Siehe Kapitel „In.Cp.Settings (→ 🗎 99)”
BackToPath	MC_KIN_IN_CP_BACKTOPATH	Siehe Kapitel „In.Cp.BackToPath (→ 🗎 99)”
Segment	MC_KIN_IN_CP_SEGMENT	Siehe Kapitel „In.Cp.Segment (→ 🗎 100)”
Path	MC_KIN_IN_CP_PATH	Siehe Kapitel „In.Cp.Path (→ 🗎 104)”

20211732 / DE – 07/2014

11.7.1 In.Cp.Settings

Typ: MC_KIN_IN_CP_SETTINGS (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
PlusTargetCartABC	BOOL	Wenn TRUE im Zyklus des Wechsels (von einer anderen Betriebsart als ContinuousPath) in ContinuousPath hinein, werden die ABC-Orientierungskoordinaten nicht mittels ContinousPath interpoliert, sondern mittels KIN_TARGET_CART über die Eingangsvariable In.Target, siehe Kaitel. „In.Target (→ 🗎 95)“. Diese Funktionalität ist auch im Kapitel „Kombination von ContinuousPath- und TARGET-Betriebsarten (→ 🗎 79)“ beschrieben.
IncludeABC	BOOL	Wenn TRUE im Zyklus des Wechsels (von einer anderen Betriebsart als ContinuousPath) in ContinuousPath hinein, werden die ABC-Orientierungskoordinaten synchronisiert mittels ContinousPath interpoliert. *PlusTargetCartABC* und *IncludeABC* können nicht beide gleichzeitig aktiviert sein.
IncludeAxis7	BOOL	Wenn TRUE im Zyklus des Wechsels (von einer anderen Betriebsart als ContinuousPath) in ContinuousPath hinein, wird Achse 7 synchronisiert mit ContinousPath Interpolation verfahren.
IncludeAxis8	BOOL	Wenn TRUE im Zyklus des Wechsels (von einer anderen Betriebsart als ContinuousPath) in ContinuousPath hinein, wird Achse 8 synchronisiert mit ContinousPath Interpolation verfahren.
UseMasterPosition	BOOL	Wenn TRUE im Zyklus des Wechsels (von einer anderen Betriebsart als ContinuousPath) in ContinuousPath hinein, werden durch die Kinematiksteuerung keine CP-Verfahrprofile generiert. Statt dessen muss das Positionsprofil für den Bahnfortschritt entlang der eingegebenen CP-Bahn applikativ mit stetigem Signalverlauf in der TaskPriority in der Variablen *In.MasterPosition.Cp* eingegeben werden, siehe Kapitel „In.MasterPosition (→ 🗎 94)“. **Hinweis:** Die Eingangsvariablen „In.General.Kin.FeedEnable/RapidStop“ werden in diesem Fall NICHT verwendet. Verlassen des Arbeitsraums sowie Drehzahlüberschreitung sowie andere Fehler führen NICHT zum Abbremsen der Bewegung.

11.7.2 In.Cp.BackToPath

Typ: MC_KIN_IN_CP_BACKTOPATH (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
Enable	BOOL	Aktivierung der Rück-Positionierung auf den CP-Pfad, z. B. nach einem Umrichterfehler. Statusinformation über die Rück-Positionierung kann der Variablen Out.Cp.BackToPath entnommen werden, siehe Kapitel „Out.Cp (→ 🗎 112)“. **Hinweis:** „In.Target.Axis“ muss mit den gewünschten kinematischen Bewegungsgrößen belegt werden, da diese für die Rück-Positonierung verwendet werden. Die Rück-Positonierung ist nur dann möglich, wenn seit Auftreten des Fehlers, der zum Verlassen der CP-Bahn führte, keine andere Betriebsart als ContinuousPath angelegt wurde.

20211732 / DE – 07/2014

Variable	Typ	Bedeutung
FeedEnable	BOOL	Vorschubfreigabe für die Rück-Positionierung auf den Pfad. Diese Variable kann während der Rück-Positionierung gesetzt und zurückgesetzt werden.
MaxInitialDistance	LREAL	Maximaler Abstand (translatorischer Abstand in [Anwendereinheit]) von der Bahn. Ist dieser Wert zu Beginn der Rück-Positonierung überschritten, wird ein Fehler generiert, dessen Message den tatsächlichen Abstand anzeigt. In dem Fall kann nach Anlegen eines ausreichend großen Werts erneut versucht werden, die Rück-Positionierung zu starten.

11.7.3 In.Cp.Segment

Typ: MC_KIN_IN_CP_SEGMENT (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
ID	INT	ID für die eindeutige Identifikation des ContinuousPath Segments und als Trigger für die Übernahme eines neuens CP-Segments in die Kinematiksteuerung In.Cp.Segment.ID <> Out.Cp.Queue.LastMappedSegID signalisiert der Kinematiksteuerung, dass ein neues Segment angelegt ist, welches in die Steuerung übernommen werden soll. Beim Abfahren der ContinuousPath Bahn wird die ID des jeweils aktuellen Segments in der Ausgangsvariablen Out.Cp.Queue.ActSegID angezeigt.
UserID	DWORD	Anwender ID des CP-Segments OHNE Funktion für die Kinematiksteuerung Beim Abfahren der ContinuousPath Bahn wird die ID des jeweils aktuellen Segments in der Ausgangsvariablen Out.Cp.Queue.ActUserID angezeigt.
Mode	MC_KIN_MODE	Gewöhnlich wird die Betriebsart in der Variablen *In.General.Mode* eingestellt. Alternativ kann dort statt einer konkreten ContinuousPath-Betriebsart (z. B. KIN_LIN_XY, KIN_CIRC_ZX, …) auch die übergreifende Betriebsart KIN_CONTINUOUS_PATH angelegt werden. In dem Fall muss die konkrete ContinuousPath-Betriebsart hier in der Variablen In.Cp.Segment.Mode eingegeben werden. Der Vorteil liegt darin, dass die CP-Segmente des Datentyps MC_KIN_IN_CP_SEGMENT mit der kompletten Segmentbeschreibung inklusive dem konkreten CP-Mode im Applikationsprogramm z. B. aus einem Array von CP-Segmenten einfach komplett auf den Eingang *In.Cp.Segment* der Kinematiksteuerung kopiert werden können.
Position	ARRAY[1..8] OF LREAL	Zielposition des CP-Segments im aktuellen Koordinatensystem (AxisGroupKin.Out.General.Kin.CoordSys)

20211732 / DE – 07/2014

Variable	Typ	Bedeutung	
Circ	MC_KIN_IN_CP_SEG-MENT_CIRC	Zusätzliche Kreisparameter Mode: MC_KIN_CIRC_MODE	
		KIN_CENTER_ANGLE • Eingabe des Kreismittelpunkts in *AuxPos* • Eingabe des zu überstreichenden Winkels in *Angle*. **Hinweis:** *Radius* und *Segment.Position* werden **NICHT** verwendet.	
		KIN_CENTER_ENDPOS • Eingabe des Kreisendpunkts in *Segment.Position* • Eingabe des Kreismittelpunkts in *AuxPos* **Hinweise:** • *Radius* und *Angle* werden **NICHT** verwendet. • Es erfolgt eine automatische Kreismittelpunktskorrektur. • Die maximal zulässige Korrektur ist konfiguriert: Config.Cp.CircCenterCorrectionThreshold, Standardwert 100000	
		KIN_INTERPOS_ENDPOS • Eingabe des Kreisendpunkts in *Segment.Position* • Eingabe eines zusätzlichen Punkts auf dem Kreisbogen in *AuxPos* **Hinweis:** • *Radius* und *Angle* werden **NICHT** verwendet.	
		KIN_RADIUS_ANGLE • Eingabe des Kreisradius in *Radius* • Eingabe des zu überstreichenden Winkels in *Angle* , **Hinweise:** • *AuxPos* und *Segment.Position* werden **NICHT** verwendet. • Das Kreisssegment wird tangential angehängt. **Voraussetzung:** Die Variable *Out.Cp.Path.EndTangentValid* muss gesetzt sein.	
		KIN_RADIUS_ENDPOS • Eingabe des Kreisradius in Radius • Eingabe des Kreisendpunkts in *Segment.Position* **Hinweise:** • *AuxPos* und Angle werden **NICHT** verwendet. • Wenn der Radius negiert eingegeben wird, wird der längere der beiden möglichen Bögen zum Kreisendpunkt ausgewählt. Bei positiver Eingabe des Radius wird der kürzere Bogen ausgewählt.	
		Direction	MC_KIN_DIRECTION In der Draufsicht auf die XY, YZ bzw. ZX-Ebene • KIN_CW: im Uhrzeigersinn • KIN_CCW: gegen den Uhrzeigersinn

Variable	Typ	Bedeutung		
			Hinweise: • Direction wird **NICHT** verwendet, wenn Mode = KIN_INTERPOS_ENDPOS.	
		AuxPos	ARRAY[1..6] OF LREAL	z. B. Kreismittelpunkt, Elemente 4..6 reserviert
		Radius	LREAL	[Anwendereinheit]
		Angle	LREAL	Vom Kreisbogen zu überstreichender Winkel, konfigurierte [rotatorische Anwendereinheit] = [Degr] oder [Rad],> 0, theoretisch beliebig viele Kreisumläufe eingebbar durch einen Winkel > 360° (Begrenzung nur durch Gesamtpfadlänge, welche die Grenze der mit 64 Bit Floating-Point Arithmetik maximal darstellbaren, fein genug gerasterten Werte nicht überschreiten darf (> 1000.000.000.000.000.000.000… [Anwendereinheiten]); wenn EllipticDistortionFactor <> 1.0, wird der Wert von Angle automatisch aufgerundet auf ganzzahlige Vielfache von PI (bzw. 180degr).
		EllipticDistortion-Factor	LREAL	Elliptische Verzerrung des Kreisbogens > 0, entspricht Faktor zur Streckung oder Verkürzung der Kreisachse parallel zur Start-Tangente. Die Pfadlänge und Geschwindigkeit wird für den unverzerrten Kreis bestimmt (EllipticDistortionFactor = 1.0), d.h. die Variable Out.Cp.Path.Distance zeigt die restliche Pfadlänge für EllipticDistortionFactor = 1.0 an, obwohl <> 1 eingestellt ist. → Distortion-Werte <> 1.0 führen zu Verzerrungen der Geometrie sowie des Geschwindigkeitsprofils
Blending	MC_KIN_IN_CP_SEGMENT_BLENDING	Parameter für das Überschleifen ZU diesem Segment. Das heißt, sie werden NICHT verwendet, wenn die Cp.Queue leer ist, also Out.Cp.Queue.Size = 0.		
		Distance	LREAL	Max. Überschleif-Abstand, >= 0.0

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
		LimitationPercentage	UINT	[%], verwendet zur Begrenzung der Blending.Distance in DIESEM CP-Segment auf 1..99% der Länge von DIESEM Segment. Das heißt, die Blending.Distance wird automatisch auf die LimitationPercentage des auf den Überschleifbogen direkt folgenden Segments begrenzt.
		Velocity.Profile	MC_KIN_CP_BLENDING_VELOCITY_PROFILE • KIN_HANDLING: Für Blending bei Handlingbahnen • KIN_LOWER: Kleinere der angrenzenden Bahngeschwindigkeit • KIN_UPPER: Größere der angrenzenden Bahngeschwindigkeit • KIN_PRECEDENT: Vorhergehende Bahngeschwindigkeit • KIN_SUBSEQUENT: Bahngeschwindigkeit im aktuellen Segment • KIN_AVERAGE: Durchschnitt der vorherg./akt. Bahng. • KIN_CENTRIFUGAL: Begrenzung der Zentrifugalkräfte	
		Velocity.Percentage	UINT 1 .. konfigurierter Maximalwert in [%], Defaultwert 100% Prozentuale Skalierung des Velocity.Profiles	
		• **Hinweis:** Bei Einstellung von *Velocity.Profile := KIN_ CENTRIFUGAL* wird das Minimum der Translation.Deceleration des vorhergenden CP-Segments und der Translation.Acceleration des aktuell eingegebenen CP-Segments zur Begrenzung der Zentrifugalbeschleunigung im Überschleifbogen verwendet. Sollte im Überschleifbogen trotzdem etwas schneller verfahren werden, können Sie eine Velocity.Percentage > 100% einstellen. Hierzu müssen Sie die *Config.Cp.MaxBlendingVelocityPercentage* in der MotionStudio-Berechtigungsstufe „Advanced“ auf der Expertenseite der Konfiguration auf einen Wert > 100 einstellen. Die Bahngeschwindigkeit im Überschleifbogen wird zusätzlich auf den Durchschnittswert der Translation.Velocity des vorhergehenden und des aktuell eingegeben Segments begrenzt. • **Hinweis:** Überschleifen wird spezifiziert mit dem CpSegment, ZU welchem der Überschleifbogen hinführt. Es ist auf diese Weise realisiert, da sich der nächste Zielpunkt bei dynamischen Prozessen oft erst während der Bewegung ergibt. Zeitlich gesehen können i. a. erst mit dem neuen Zielpunkt die Art und die Bewegungsgrößen der Interpolation zum Zielpunkt sowie die Art und die Bewegungsgrößen des Überschleifbogens AUF das Segment zum Zielpunkt eingegeben werden. Deshalb kann das Überschleifen noch nicht mit dem letzten Segment parametriert werden.		

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
Translation	MC_KIN_IN_CP_SEGMENT_TRANSLATION	In dieser Struktur werden die kinematischen Bewegungsgrößen für die Translation des TCP entlang der ContinuousPath Bahn angegeben. Beispielsweise entspricht die *Translation.Velocity* der maximalen Bahngeschwindigkeit in diesem Segment.		
		Velocity	LREAL	[Anwendereinheit / sec], > 0
		Acceleration	LREAL	[Anwendereinheit / (sec * sec)], > 0
		Deceleration	LREAL	Anwendereinheit / (sec * sec)], > 0, <= *Config.Cp.RapidTransDeceleration*
		Jerk	LREAL	[ms], >= Config.Cp.RapidTrans-Jerk
Cart	MC_KIN_IN_CP_SEGMENT_CART	In den Array-Komponenten 4..6 spezifizieren Sie die Bewegungsparameter (Vel, Acc, Dec, Jerk) für die synchronisierte Rotation (ABC) des Werkzeugs. Wird nur verwendet, wenn Out.Cp.Settings.IncludingABC = TRUE. Die Array-Komponenten 1..3 sind reserviert.		
Axis	MC_KIN_IN_CP_SEGMENT_AXIS	In den Array-Komponenten 7, 8 spezifizieren Sie die Bewegungsparameter (Vel, Acc, Dec, Jerk) für die synchronisierte Bewegung der Hilfsachsen (A7, A8). Wird nur verwendet, wenn Out.Cp.Settings.IncludingA7 bzw. ..A8 = TRUE. Die Array-Komponenten 1..6 sind reserviert.		

11.7.4 In.Cp.Path

Typ: MC_KIN_IN_CP_PATH (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
VelocityPercentage	UINT	0 .. konfigurierter Maximalwert in [%], Defaultwert 100% Prozentuale Skalierung des programmierten Geschwindigkeitsprofils, kann während Bahnbewegung geändert werden.
Reverse	BOOL	TRUE: Die Bahn wird rückwärts abgefahren, maximal bis zum Erreichen des ersten Queue-Eintrags, nachdem die Queue leergelaufen war und der noch nicht durch kontinuierliche Befüllung der Queue durch neue Elemente im Ringpuffer-Prinzip überschrieben wurde.
FidelityPercentage	USINT	90% bis 100% 100%: Exakte Bahnsteuerung <100%: Glättung von Krümmungssprüngen **Hinweis:** Wird beim Wechsel (von einer anderen Betriebsart als ContinuousPath) in ContinuousPath sowie bei “Stillstand im aktuellen Koordinatensystem” übernommen. Bei kleinen Override-Werten kann sich aufgrund Ressourcen-Begrenzungen eine erhöhte Bahngenauigkeit ergeben gegenüber der Einstellung Override = 100%.

20211732 / DE – 07/2014

11.8 In.Simu3D

Typ: ST_AxisGroupKin_In_Simu3D (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)

Variable	Typ	Bedeutung		
Enable	BOOL	Verbindung zum 3D Simulations-PC		
Ctrl	MC_KIN_SIMU3D_CONTROL_AGK (04_InterpolatedMotion\ MPLCSimu3D.lib)	Pen_TCP_KCS	BOOL	Stift am TCP, Linien relativ zum KCS
		Pen_TCP_WCS	BOOL	Stift am TCP, Linien relativ zum WCS
		Pen_TCP_PCS 1	BOOL	Stift am TCP, Linien relativ zum PCS1
		Pen_TCP_PCS 2	BOOL	Stift am TCP, Linien relativ zum PCS2
		DeleteAllPenLines	BOOL	Löschen aller Linien

20211732 / DE – 07/2014

11.9 In.Diag

Typ: ST_AxisGroupKin_In_Diag (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)

Hier haben Sie die Möglichkeit, die Ausgabe nützlicher Messages für verschiedene Funktionalitäten zu aktivieren.

Variable	Typ	Bedeutung
Kin.MessageAtAutoSlowdown	• MotorSpeed: BOOL • Centrifugal: BOOL • CartRotation: BOOL • AuxAxis: BOOL	Message bei automatischer Verringerung der Bahngeschwindigkeit
Config.MessageAtReadingXML	BOOL	Message beim Einlesen von XML-Dateien
Simu3D.MessageAtCommunicationError	BOOL	Message bei Kommunikationsproblemen zur 3D-Simulation

20211732 / DE – 07/2014

11.10 Out.General

Typ: ST_AxisGroupKin_Out_General (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)

Variable	Typ	Bedeutung	
Inverter	ST_AxisGroupKin_Out_General_Inverter (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)	Connected	ARRAY[0..8] OF BOOL
		Referenced	ARRAY[0..8] OF BOOL
		Powered	ARRAY[0..8] OF BOOL
		Ready	ARRAY[0..8] OF BOOL
		Error	ARRAY[0..8] OF BOOL
		InGear	ARRAY[0..8] OF BOOL
		InKinMode	ARRAY[0..8] OF BOOL
		StandStill	ARRAY[0..8] OF BOOL
		SafeTorqueOff	ARRAY[0..8] OF BOOL
		HWLS	ARRAY[0..8] OF BOOL
		Simulation	ARRAY[0..8] OF BOOL
		State_Error	ARRAY[1..8] OF BYTE
		AxisControlError	ARRAY[0..8] OF BOOL
		Die Bits 1..8 beziehen sich auf die Achsen 1..8. Bit 0 bezieht sich auf alle konfigurierten Achsen dieser Kinematikinstanz.	
Axis	MC_KIN_OUT_GENERAL_AXIS (04_InterpolatedMotion \MPLCKinematics.lib)	Motor-Drehzahlüberwachung	
		MotorSpeed	ARRAY[1..8] OF LREAL
		PastAbsoluteReachedMotorSpeed	ARRAY[1..8] OF LREAL
		Hinweis: PastAbsoluteReachedMotorSpeed wird bei steigender Flanke am Eingang In.General.Reset sowie beim Wechsel in eine Kinematik-Betriebsart (z. B. nachdem eine Achse nicht betriebsbereit war) sowie im Mode KIN_AM_DEFAULT zurückgesetzt.	

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
Kin	ST_AxisGroupKin_Out_General_Kin (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)	CoordSys	MC_KIN_COORDSYS	
		Constellation	UINT;	Kinematik Konstellation z. B. bei SCARA Ellbogen links/rechts
		LimitValue	ARRAY[1..12] OF LREAL	Werte der Kinematiklimitierungen (z. B. Winkel)
		ChainItem	ARRAY[1..24] OF LREAL	Detaildaten der kinematischen Kette, z. B. Ellbogen-Position, in Bezug auf BCS,abhängig vom konfigurierten Kinematikmodell
		TranslationVelocity	LREAL;	Translationsgeschwindigkeit des TCP in KCS, wenn CoordSys = KCS, ACS, TCS, ansonsten im aktuellen “CoordSys”
		OverridePercentage	UINT;	
		Ready	BOOL;	TRUE: Bereit für kinematische Betriebsarten
		Done	BOOL;	TRUE: Der aktuelle Kinematik-Bewegungsauftrag ist komplett ausgeführt.

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
Position	MC_KIN_OUT_GENERAL_POSITION (04_InterpolatedMotion\MPLCKinematics.lib)	ActCoordSys	ARRAY[1..8] OF LREAL	Position im aktuellen Koordinatensystem in KCS, wenn CoordSys = TCS , ansonsten im aktuellen “CoordSys”
		ACS	ARRAY[1..8] OF LREAL	Position in ACS
		ACS_Incr	:ARRAY[1..8] OF DINT	Position in ACS Inkrementen
		KCS	ARRAY[1..8] OF LREAL	Position in KCS
		WCS	ARRAY[1..8] OF LREAL	Position in WCS
		PCS1	ARRAY[1..8] OF LREAL	Position in PCS1
		PCS2	ARRAY[1..8] OF LREAL	Position in PCS2
		KCS_WCS_PCS_Valid	BOOL	FALSE: Kartesische Position bzw. Orientierung nicht gültig, z. B. da nicht erreichbar; oder da mindestens eine der verwendeten Achsen 1..6 nicht referenziert ist. („Verwendet“ werden die Achsen 1 bis *Config.General.NumberOfKinAxes*, bei denen *In.General.Ignore.Inverter[n]* = FALSE, z. B. *NumberOfKinAxes* = 4 bei KIN_SCARA_RRRL_XYZA_M10)
SWLS	MC_KIN_OUT_GENERAL_SWLS (04_InterpolatedMotion\MPLCKinematics.lib)	Arbeitsbereichsüberwachung: Achsen / kartesisch / kinematisch		
		Axis	AtLeastOn	BOOL
			Positive	ARRAY[1..8] OF BOOL;
			Negative	ARRAY[1..8] OF BOOL;
		Cart	AtLeastOne	BOOL
			Positive	ARRAY[1..6] OF BOOL;
			Negative	ARRAY[1..6] OF BOOL;

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
		Kin	AtLeastOne	BOOL
			Positive	ARRAY[1..12] OF BOOL;
			Negative	ARRAY[1..12] OF BOOL;
		Past_Axis_Cart_Kin		BOOL
		MotorSpeedWarning		ARRAY[0..8] OF BOOL;
		PastMotorSpeedWarning		ARRAY[0..8] OF BOOL
		MotorSpeed		ARRAY[0..8] OF BOOL
		PastMotorSpeed		ARRAY[0..8] OF BOOL Wenn Bit 0 gesetzt ist, liegt eine entsprechende Überschreitung bei mindestens einer Achse vor.
		Hinweis: Past_Axis_Cart_Kin, PastMotorSpeedWarning und PastMotorSpeed wird bei steigender Flanke am Eingang *In.General.Reset* sowie beim Wechsel in eine Kinematik-Betriebsart (z. B. nachdem eine Achse nicht betriebsbereit war) sowie im Mode KIN_AM_DEFAULT zurückgesetzt.		

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
Standstill	MC_KIN_OUT_GENE-RAL_STAND-STILL (04_Inter-polatedMotion \MPLCKinema-tics.lib)	ActCoordSys	BOOL	Stillstand im aktuellen Koordina-tensystem in KCS, wenn CoordSys = TCS, ansonsten im aktuellen “Coord-Sys” umfasst Stillstand der Hilfsachsen A7/8
		ACS	ARRAY[0..8] OF BOOL	Stillstand der Achsen Die Bits 1..8 beziehen sich auf die Achsen 1..8. Bit 0 bezieht sich auf alle konfigu-rierten Achsen dieser Kinematikin-stanz.
		KCS_TCP	BOOL	Stillstand des TCP im KCS
		WCS_TCP	BOOL	Stillstand des TCP im WCS
		PCS1_TCP	BOOL	Stillstand des TCP im PCS1
		PCS2_TCP	BOOL	Stillstand des TCP im PCS2
Mode	MC_KIN_MODE	Aktuelle Betriebsart der Kinematikinstanz: KIN_AM_DEFAULT, KIN_AM_HOMING, KIN_STOP, KIN_JOG_AXIS, KIN_JOG_CART, KIN_TARGET_AXIS, KIN_TARGET_CART, KIN_LIN_XY, KIN_LIN_YZ, KIN_LIN_ZX, KIN_LIN_3D, KIN_CIRC_XY, KIN_CIRC_YZ, KIN_CIRC_ZX, KIN_CP_BLENDING		
ModeType	MC_KIN_MODE	Typ der aktuellen Betriebsart: KIN_AM_DEFAULT, KIN_AM_HOMING, KIN_STOP, KIN_JOG_AXIS, KIN_JOG_CART, KIN_TARGET_AXIS, KIN_TARGET_CART, KIN_CONTINUOUS_PATH		
FBError	BOOL	TRUE: In der Kinematikinstanz liegt ein Fehler vor. • TRUE und AxisGroupKin.Out.General.Ready = TRUE Die Bewegung wird im aktuellen Koordinatensystem abgebremst. **ACHTUNG:** In einer kartesischen Betriebsart (z. B. KIN_TAR-GET_CART, KIN_LIN_3D) z. B. in einem bewegten Koordinaten-system PCS wird die Bewegung des Tools relativ zum aktuellen, bewegten PCS abgebremst. Die Achsen bewegen sich i. a. noch. Wenn in dem Fall die Achsen zum Stillstand kommen sollen, muss		

20211732 / DE – 07/2014

Variable	Typ	Bedeutung
		applikativ in Interpolation der Achsen (z. B. KIN_JOG/ TARGET_AXIS) umgeschaltet werden. Siehe auch Hinweis in Kap. „In.General (→ 🗎 90)“, Eingangsvariable Kin.FeedEnable/RapidStop. • TRUE und AxisGroupKin.Out.General.Ready = FALSE Die Bewegung wird NICHT-kinematisch achsweise durch MultiMotion Profilgeneratoren abgebremst. FALSE: Es liegt kein Fehler vor.
FBErrorID	DWORD	Siehe Kap. “Fehlercodes (→ 🗎 148)”

11.11 Out.Homing

Typ: ST_AxisGroupKin_Out_Homing (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)

Variable	Typ	Bedeutung
Done	ARRAY[0..8] OF BOOL	Referenzieren erfolgreich durchgeführt. Die Bits 1..8 beziehen sich auf die Achsen 1..8. Bit 0 bezieht sich auf alle konfigurierten Achsen dieser Kinematikinstanz.

11.12 Out.Cp

Typ: MC_KIN_OUT_CP (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung		
Settings	MC_KIN_OUT_CP_SETTINGS (04_InterpolatedMotion \MPLCKinematics.lib)	ContinuousPath Einstellungen, siehe auch Kap. “In.Cp.Settings (→ 🗎 99)”.		
		PlusTargetCartABC	BOOL	
		IncludingABC	BOOL	
		IncludingAxis7	BOOL	
		IncludingAxis8	BOOL	
		MasterPosUse	BOOL	
BackToPath	MC_KIN_OUT_CP_BACKTOPATH (04_InterpolatedMotion \MPLCKinematics.lib)	BackToPath Zustand, siehe auch Kap. „In.Cp.BackToPath (→ 🗎 99)“.		
		InitialDistance	LREAL	Translatorischer Abstand in [Anwendereinheit]) von der Bahn zu Beginn der Rück-Positonierung.

20211732 / DE – 07/2014

Variable	Typ	Bedeutung		
		Active	BOOL	TRUE: Die Rück-Positionierung auf den Pfad ist im Gange. (Auch TRUE, wenn die Achsen sich nicht bewegen wegen *In.Cp.BackToPath.FeedEnable* = FALSE oder *In.Target.Axis.Velocity/Acceleration[i]* = 0)
		Done	BOOL	TRUE: Die Rück-Positionierung auf den Pfad ist abgeschlossen, z. B. nach einem Umrichterfehler. „Done“ wird mit *In.Cp.BackToPath.Enable* zurückgesetzt.
Queue	MC_KIN_OUT_CP_QUEUE	LastMappedSegID	INT	ID des letzten CP-Segments, das erfolgreich in die Queue übernommen wurde, unabhängig davon, ob dieses Segment noch in der Queue ist oder bereits komplett abgefahren wurde.
		Hinweis: In einer Schrittkette zur Befüllung der CP-Queue kann z. B. die Ausgangsvariable *Out.Cp.Queue.LastMappedSegID* einfach inkrementiert werden, um eine geeignete ID (Eingangsvariable *In.Cp.Segment.ID*) für das nächste zu befüllende CP-Segment zu erhalten.		
		ActSegID	INT	Zeigt die Segment.ID des Segments an, das gerade durchfahren wird.
		ActUserID	DWORD	Zeigt die Segment.UserID des Segments an, das gerade durchfahren wird.
		Size	UINT	Anzahl der CP-Segmente (OHNE die durch die Kinematiksteuerung hinzugefügten Blending-Segmente), die sich gerade in der Queue befinden und noch nicht komplett durchfahren wurden.
		SizeWithBlending	UINT	Anzahl der CP-Segmente, die sich gerade in der Queue befinden und noch nicht komplett durchfahren wurden.
		Hinweis: Ein automatisch durch die Kinematiksteuerung eingefügtes Blending-Segment hat die gleiche Segment.ID/UserID wie das CP-Segment, ZU welchem es überschleift. Die Variable Out.Cp.Queue.SizeWithBlending umfasst auch die automatisch eingefügten Blending-Segmente.		

Variable	Typ	Bedeutung		
Path	MC_KIN_OUT_CP_PATH	Distance	LREAL	Rest-Bahnlänge bis zum Erreichen des Zielpunkts des zuletzt übernommenen CP-Segments.
		EndReached	BOOL	TRUE: CP-Bahnende erreicht. Kann verschieden sein von (*Cp.Queue.SizeWithBlending* = 0) oder (*Cp.Path.Distance* = 0.0), wenn z. B. (In.Cp.Path.FidelityPercentage < 100%)
		EndTangentValid	BOOL	TRUE: Die Tangente des zuletzt eingegebenen CP-Segments ist gültig. Somit ist die relative Eingabe von CP-Segmenten möglich. (z. B. KIN_CIRC_xx mittels KIN_RADIUS_ANGLE) FALSE: Die relative Eingabe . (z. B. KIN_CIRC_xx mittels KIN_RADIUS_ANGLE) ist nicht möglich.

11.13 Out.Simu3D

Typ: ST_AxisGroupKin_Out_Simu3D (03_ApplicationModules\ MPLCAxisGroupControl_Kinematics.lib)

Variable	Typ	Bedeutung
Dummy	BOOL	Reserviert

11.14 Out.Diag

Typ: MC_KIN_OUT_DIAG (04_InterpolatedMotion\MPLCKinematics.lib)

Variable	Typ	Bedeutung
NoAttemptToConfigure	BOOL	TRUE, solange seit dem Start der MOVI-PLC® noch nicht versucht wurde, die Kinematiksteuerung zu konfigurieren, z. B. da die erforderlichen Achsen nicht verbunden sind.
NewConfigTakenOver	BOOL	Reserviert
NumberOfSuccessfulConfigurations	UINT	Wird einmal inkrementiert beim Wechsel (von einer NICHT-kinematischen Betriebsart, z. B. KIN_AM_DEFAULT) in eine kinematische Betriebsart (z. B. KIN_JOG_AXIS) sowie zyklisch in der Betriebsart KIN_AM_DEFAULT unter Verwendung der von den Umrichtern gelesenen Achsinkrementen.
KinModel	MC_KIN_MODEL (04_InterpolatedMotion\MPLCKinematics.lib)	Kinematikmodelle (→ 🗎 57)
UserKinNum	INT	Nummer der User-Kinematik, wenn KinModel = KIN_USER_KINEMATICS

20211732 / DE – 07/2014

Variable	Typ	Bedeutung	
MaxAllowedAxisVelocity	ARRAY[1..8] OF LREAL	Zeigt die maximal zulässige Achsen-Geschwindigkeiten in [Anwendereinheit / sec] an, abgeleitet von der jeweils maximalen Motordrehzahl und WarningPercentage. Diese Werte sind die maximal zulässigen Geschwindigkeiten (In.Target.Axis.Velocity) für die Betriebsart KIN_TARGET_AXIS.	
LastComputationTime	DWORD	Rechenzeit in [usec] der Kinematik-Bahnberechnung im letzten Zyklus der TaskPriority.	
MaxComputationTime	DWORD	Maximale Rechenzeit in [usec] der Kinematik-Bahnberechnung seit der letzten Konfiguration der Kinematiksteuerung (Wechsel in eine Kinematik-Betriebsart, z. B. nachdem eine Achse nicht betriebsbereit war, sowie zyklisch im Mode KIN_AM_DEFAULT)	
License	MC_KIN_OUT_DIAG_LICENSE(04_InterpolatedMotion\MPLCKinematics.lib)	Aktivierte Lizenz sowie die hierzu konsumierten Technologiepunkte für diese Kinematikinstanz (siehe Erforderliche Technologiestufe (→ 🗎 13)	
		ConsumedTechnologyCredits	UINT
		Mode_Jog_Target	BOOL
		Mode_Lin2D_Circ2D	BOOL
		Mode_Lin3D_Circ3D_AxisCP	BOOL
		Model_No_CartesianGantry	BOOL
		World_Piece_CoordinateSystem	BOOL

20211732 / DE – 07/2014

12 Bausteine der Bibliothek MPLCKinematics

Die Bibliothek `04_InterpolatedMotion/MPLCKinematics.lib` enthält nützliche Bausteine für die Erstellung des Anwenderprogramms.

- MC_KinInProximity
- MC_KinMonitorCpQueueSegment
- MC_KinCoordSysMeasurement
- MC_KinEncoderDataProcessing
- MC_KinWcsPcs1Assignment
- MC_KinWcsPcs2Assignment
- MC_KinProfGen

12.1 Funktion MC_KinInProximity

Die Funktion MC_KinInProximity kann im Bewegungsablauf-Programm für die Prüfung genutzt werden, ob die Entfernung zur Zielposition oder zur Zielorientierung maximal einen parametrierbaren Grenzwert aufweist.

Bei TARGET Interpolation können Sie diese Funktion innerhalb einer Ablauf-Schrittkette in den Transitionen zum Weiterschalten in den nächsten Schritt und somit zur neuen Zielvorgabe nutzen.

Auch bei der ContinuousPath Interpolation können Sie entsprechend eine Schrittkette aufbauen, wenn das nächste ContinuousPath Segment erst dann in die Queue eingetragen werden soll, wenn z. B. die Entfernung zur Zielorientierung einen bestimmten Wert unterschreitet. Auf diese Weise ist die Koordination von ContinuousPath Segmenten und zugehörigen Orientierungswerten realisierbar. Beispielsweise wird in einer Pick'n'Place Bewegung das Segment zum Absenken des Greifers erst dann in die Kinematiksteuerung eingespeist, wenn die mittels TARGET Interpolation angesteuerte Drehung des Greifers im Raum den erforderlichen Winkel erreicht hat (Kapitel „Kombination von ContinuousPath- und TARGET-Betriebsarten (→ 🗎 79)“).

Eingangssignal	Typ	Bedeutung
Threshold	LREAL	Grenzwert für die Prüfung der Entfernung (Winkel oder translatorischer Abstand, abhängig von den Einstellungen in *CoordSys* und *Dimension*).
Position	ARRAY[1..8] OF LREAL	Achswerte oder Pose, zu der die Entfernung geprüft wird.
CoordSys	MC_KIN_COORDSYS	Koordinatensystem, in dem die Prüfung der Entfernung stattfinden soll. **Hinweis:** In der Einstellung KIN_TCS wird automatisch im KCS geprüft.

20211732 / DE – 07/2014

Eingangssignal	Typ	Bedeutung
Dimension	BYTE	Im Byte *Dimension* erfolgt die bitweise Einstellung, welche Dimensionen für die Bestimmung der Entfernung einbezogen werden sollen, z. B.: 2#0000_0011: Dimensionen 1, 2 2#0011_1000: Dimensionen 4, 5, 6 **Hinweis:** Die Entfernungsprüfung in einem kartesischen Koordinatensystem (Eingangssignal *CoordSys*; KIN_KCS/WCS/PCS1/2) verwendet in den Dimensionen 4/5/6 die ABC-Orientierungswerte. Diese sind jedoch mehrdeutig. Bei Verwendung von KIN_TARGET_AXIS und Angabe des Ziels in kartesischen Koordinaten fährt die Kinematik das Ziel mit den nächsten Achs-Tupel an. Die ABC-Werte werden dabei nach Möglichkeit stetig nachgeführt. Es kann passieren, dass die Zielachswerte erreicht sind, aber mit einer anderen ABC-Darstellung als im Eingangssignal *Position* angelegt ist. In diesem Fall schaltet die Funktion den Rückgabewert (bei kleinem *Threshold*) nicht auf den Wert TRUE.

Rückgabewert	Typ	Bedeutung
MC_KinInProximity	BOOL	TRUE, wenn die Entfernung zum Ziel (Eingangssignal *Position*), das im angegebenen Koordinatensystem (Eingangssignal *CoordSys*) interpretiert wird, maximal den Wert Threshold aufweist.

Ein-/Ausgangssignal	Typ	Bedeutung
ActPos	MC_KIN_OUT_GENERAL_POSITION	Aktuelle Position, verwendet zur Bestimmung des Abstands zum Eingangssignal *Position*.

12.2 Funktionsbaustein MC_KinCoordSysMeasurement

Sie können den Funktionsbaustein MC_KinCoordSysMeasurement dazu verwenden, ein Koordinatensystem zu vermessen, z. B. die Position und Orientierung eines Transportbands im WCS-Koordinatensystem.

Dazu müssen nacheinander 3 Punkte des zu vermessenden Koordinatensystems aus Sicht des Bezugs-Koordinatensystems (z. B. WCS) am Eingangssignal Position angelegt werden. Bei diesen Punkten handelt es sich um den Ursprung, einen Punkt auf der positiven X-Achse sowie einen Punkt in der XY-Ebene (positive Y-Koordinate) des zu vermessenden Koordinatensystems.

Liegt der jeweils einzugebende Punkt am Eingang *Position* an, müssen Sie jeweils eine steigende Flanke am entsprechenden Eingang SetPosition_... erzeugen, um den Punkt in den Funktionsbaustein zu übernehmen. Eine steigende Flanke am Eingang *ComputeTransform* führt zur Berechnung der Transformation, die in den Ausgangssignalen *TransformVector / TransformFrame* ausgegeben wird, sobald das Ausgangssignal *Done* auf TRUE wechselt.

20211732 / DE – 07/2014

Die 3 einzugebenden Punkte können Sie z. B. dadurch bestimmen, dass Sie den TCP (z. B. einer Messspitze) im Tippbetrieb zu den 3 Punkten verfahren. Dabei sollte z. B. die Variable *AxisGroupKin.Inst[..].Out.General.Position.xCS* am Eingang *Position* angelegt sein, um die Vermessung in Bezug auf das Koordinatensystem xCS vorzunehmen. Weitere Informationen entnehmen Sie dem Kapitel „Vermessen eines Koordinatensystems (→ 🗎 130)“.

Eingangssignal	Typ	Bedeutung
EN	BOOL	TRUE: Der Funktionsbaustein wird ausgeführt FALSE: Der Funktionsbaustein wird nicht ausgeführt
Start	BOOL	TRUE: Vermessung des Koordinatensystems aktiv FALSE: Reset der Vermessung des Koordinatensystems und aller Ausgangsvariablen.
ResetError	BOOL	Bei steigender Flanke wird ein Fehler zurückgesetzt **Hinweis:** Bei einem Fehler-Reset wird die bis dahin vorgenommene Vermessung nicht verworfen.
SetPosition_CoordSysOrigin	BOOL	Bei steigender Flanke dieses Eingangssignals wird der im Eingangssignal *Position* anliegende Punkt als Ursprung des zu vermessenden Koordinatensystems übernommen.
SetPosition_AlongX_Xpositive	BOOL	Bei steigender Flanke dieses Eingangssignals wird der im Eingangssignal *Position* anliegende Punkt übernommen. Er wird interpretiert als ein Punkt auf der positiven X-Achse des zu vermessenden Koordinatensystems.
SetPosition_InXYplane_Ypositive	BOOL	Bei steigender Flanke dieses Eingangssignals wird der im Eingangssignal Position anliegende Punkt übernommen. Er wird interpretiert als ein Punkt in der XY-Ebene des zu vermessenden Koordinatensystems mit positiver Y-Koordinate.
ComputeTransform	BOOL	Bei steigender Flanke dieses Eingangssignals wird die Transformation zu dem Koordinatensystem berechnet, das vermessen werden soll.
Position	ARRAY[1..8] OF LREAL	Die ersten 3 Elemente dieses Eingangssignals werden verwendet als Punkt (X, Y, Z-Koordinaten) im Koordinatensystem, auf das sich die Transformation zu dem zu vermessenden Koordinatensystem bezieht. **Beispiel:** Wird an diesen Eingang die Variable *AxisGroupKin.Inst[..].Out.General.Position.WCS* angelegt, erfolgt die Vermessung des Koordinatensystems in Bezug auf das World Coordinate System (WCS). Das heißt, in den Ausgangssignalen *CoordSysVector / CoordSysFrame* wird die Transformation von WCS zu dem zu vermessenden Koordinatensystem ausgegeben.
TransofrmOffset	ARRAY[1..6] OF LREAL	In diesem Eingangssignal kann eine Transformation spezifiziert werden, die zusätzlich an das zu vermessende Koordinatensystem angehängt wird. Sie wird somit in die Ausgangssignale *TransformVector / TransformFrame* eingerechnet. **Hinweis:** Um keinen zusätzlichen Offset zu erzeugen, geben Sie in diesem Eingangssignal das Tupel (0,0,0,0,0,0) ein.

Ausgangssignal	Typ	Bedeutung
TransformVector	ARRAY[1..6] OF LREAL	Transformation zu dem zu vermessenden Koordinatensystem in Darstellung XYZABC.

20211732 / DE – 07/2014

Ausgangssignal	Typ	Bedeutung
TransformFrame	ARRAY[1..4, 1..4] OF LREAL	Transformation zu dem zu vermessenden Koordinatensystem in 4x4 Matrizendarstellung.
Position_CoordSysOrigin	ARRAY[1..3] OF LREAL	Anzeige des übernommenen Ursprungs des zu vermessenden Koordinatensystems.
Position_AlongX_Xpositive	ARRAY[1..3] OF LREAL	Anzeige des übernommenen Punkts auf der positiven X-Achse des zu -vermessenden Koordinatensystems.
Position_InXYplane_Ypositive	BOOL	Anzeige des übernommenen Punkts in der XY-Ebene des zu -vermessenden Koordinatensystems mit positiver Y-Koordinate.
PositionIsSet_CoordSysOrigin	BOOL	Anzeige, ob der Ursprungspunkt des zu -vermessenden Koordinatensystems übernommen wurde.
PositionIsSet_AlongX_Xpositive	BOOL	Anzeige, ob der Punkt auf der positiven X-Achse des zu -vermessenden Koordinatensystems übernommen wurde.
PositionIsSet_InXYplane_Ypositive	BOOL	Anzeige, ob der Punkt in der XY-Ebene des zu -vermessenden Koordinatensystems mit positiver Y-Koordinate übernommen wurde.
Error	BOOL	TRUE: Ein Fehler ist aufgetreten bei Vermessung des Koordinatensystems (z. B. übernommener Punkt für Ursprung und Punkt auf positiver X-Achse identisch; mehrere *SetPosition_*...-Eingangssignal gleichzeitig TRUE; nicht alle erforderlichen Punkte eingegeben).
Done	BOOL	Die Vermessung ist abgeschlossen und die Transformation steht in den Ausgangssignalen *TransformVector / TransformFrame*.

Ein-/Ausgangssignal	Typ	Bedeutung
Config	ST_AxisGroupKin_Config	Kinematik-Konfiguration der Instanz, zu deren Steuerung das Ausgangssignal *CoordSysVector* bestimmt wird.

12.3 Funktionsbaustein MC_KinEncoderDataProcessing

Der Funktionsbaustein MC_KinEncoderDataProcessing wandelt ein Gebersignal in Anwender-Positionswerte um, z. B. für das Tracking eines Objekts auf einem Transportband.

Beispielsweise können Sie das Signal einer Lichtschranke an das Eingangssignal *Trigger* anlegen. Beim Erfassen eines Triggers wird das Ausgangssignal *Position* auf den Wert des Eingangssignals *TriggerPositionOffset* gesetzt und anschließend unter Verwendung der Geberwerte aktualisiert.

Sie können das Ausgangssignal *Position* z. B. für die zyklische Aktualisierung einer Komponente (z. B. X-Koordinate) der Transformation WCS_PCSx verwenden. Zum Triggerzeitpunkt darf sich die Kinematiksteuerung nicht im Koordinatensystem PCSx befinden, da der Verlauf des Ausgangssignals Position i. a. einen Sprung aufweist. Der Funktionsbaustein MC_KinEncoderDataProcessing zeigt deswegen auch an, ob

20211732 / DE – 07/2014

die Kinematiksteuerung ausreichend Zeit hatte, die aktuelle Bewegung im neuen Koordinatensystem PCSx zu beobachten. Erst wenn der Ausgang *ObservationTimeOver* auf TRUE umgeschaltet wird, dürfen Sie den Wechsel in das neue Koordinatensystem PCSx vornehmen. Nur so wird eine sprungfreie Umschaltung gewährleistet.

HINWEIS

Der Funktionsbaustein MC_KinEncoderDataProcessing muss zyklisch in der TaskPriority aufgerufen werden. Im Bewegungsablaufprogramm, das gewöhnlich in einer anderen Task abläuft, dürfen nur die Ein- und Ausgangsvariablen der jeweiligen Instanz des MC_KinEncoderDataProcessing beschrieben oder gelesen werden.

Wenn die Geberwerte z. B. mittels SCOM Receive-Objekten von einem Umrichter gelesen werden, müssen die SCOM-Objekte mit der TaskPriority synchronisiert werden. Sie müssen in der TaskPriority dem Funktionsbaustein MC_KinEncoderDataProcessing übergeben werden.

Eingangssignal	Typ	Bedeutung
EN	BOOL	TRUE: Der Funktionsbaustein wird ausgeführt. FALSE: Der Funktionsbaustein wird nicht ausgeführt.
Standby	BOOL	Ein Trigger kann nur registriert werden, wenn das -Eingangssignal *Standby* = TRUE ist. Selbst wenn das Eingangssignal Standby = FALSE ist, werden die Ausgangssignale *Position* und *Encoder_Incr* weiter aktualisiert.
Trigger	BOOL	Bei steigender Flanke am Eingang *Trigger* wird das Ausgangssignal *Position* auf den Wert des Eingangssignals *TriggerPositionOffset* gesetzt. Auch werden die Ausgangssignale *TriggerOccurred* = TRUE und *ObservationTimeOver* = FALSE gesetzt. **Hinweis:** Solange *Standby* = TRUE ist, wird nur eine -steigende Flanke am Eingang *Trigger* erfasst. Weitere steigende Flanken werden ignoriert, um unerwartete Sprünge im Ausgangssignal Position zu vermeiden, das z. B. wiederum als Eingangssignal *AxisGroupKin.Inst[..].In.Transform.WCS_PCS1* genutzt wird. Um sicherzustellen, dass ein Trigger erfasst wird, müssen Sie *Standby* vorher auf FALSE setzen. Erst nachdem auch der Ausgang *InStandby* auf FALSE zurückgesetzt wurde, dürfen Sie Standby erneut auf TRUE setzen. Schon in dem Zyklus, in dem *Standby* auf TRUE gesetzt wird, wird der Pegel TRUE am Eingang *Trigger* als Triggersignal erfasst, unabhängig vom Pegel im vorherigen Zyklus (selbst wenn *Trigger* = TRUE war). Dieses Verhalten hat zum Ziel, einen *Trigger* garantiert erzeugen zu können. Beispielsweise können Sie eine Programmvariable mittels OR parallel zu einem Lichtschrankensignal am Eingang *Trigger* anschließen. Unabhängig vom vorherigen Verlauf des Sensorsignals führt der Wert TRUE in der Programmvariable zum Auslösen des *Triggers*. Es ist möglich, dass ein durch ein Objekt bedämpfter Sensor zum Auslösen des *Triggers* führt, in dem Zyklus, wenn *Standby* auf TRUE gesetzt wird. Dies wäre der Fall, wenn das Objekt zu dem Zeitpunkt bereits teilweise am Sensor vorbeibewegt wurde. Wenn stattdessen sicher der Moment erfasst werden soll, an dem das Objekt beginnt, den Sensor zu bedämpfen, sollten Sie einen Funktionsbaustein R_TRIG einsetzen. Legen Sie in diesem Fall den Ausgang *fb_R_TRIG.Q* an den Eingang *Trigger* an.
TriggerPositionOffset	LREAL	Position, auf welche der Ausgang *Position* beim erfassten Trigger gesetzt wird.

20211732 / DE – 07/2014

Eingangssignal	Typ	Bedeutung
EncoderWord-Low	WORD	Encoder Low Word, z. B. externes Gebersignal, das vom Umrichter mittels SCOM-Receive zyklisch in der TaskPriority eingelesen wird.
EncoderWord-High	WORD	Encoder High Word
EncoderDouble-Word	DWORD	Alternative Darstellung von Encoder Low/High Word.
UseDWord_In-steadLowHigh	BOOL	TRUE: EncoderDoubleWord wird eingelesen FALSE: EncoderWordLow/High werden eingelesen.
Numerator	LREAL	Inkrement-Delta zur Umrechnung der Geber-Inkremente in den Positionswert.
Denominator	LREAL	Positions-Delta, das dem Inkrement-Delta entspricht.
Filter	INT	In Vorbereitung

Ausgangssignal	Typ	Bedeutung
Position	LREAL	Position in Anwendereinheiten, die über die Eingangssignale *Numerator* und *Denominator* eingestellt werden Die Position wird bei einem erfassten Trigger auf den Wert des Eingangssignals *TriggerPositionOffset* gesetzt.
Encoder_Incr	DWORD	Identisch zum Eingangssignal *EncoderDoubleWord*, wenn *UseD-Word_InsteadLowHigh* =TRUE. Ansonsten aus *EncoderWordLow* und *EncoderWordHigh* zusammengesetzter Wert.
InStandby	BOOL	Rückmeldung, dass das Eingangssignal *Standby* gesetzt ist. **Hinweis:** Das Ausgangssignal wird benötigt für die Programmierung des Bewegungsablaufs in einer von der TaskPriority verschiedenen Task, um sicherstellen zu können, dass *Standby* in der TaskPriority auch tatsächlich auf FALSE zurückgesetzt wurde. Nur so kann nach dem folgenden Setzen von *Standby* sicher ein Trigger erfasst werden.
TriggerOccurred	BOOL	Rückmeldung, dass ein Trigger erfasst wurde.
ObservationTi-meOver	BOOL	Das Ausgangssignal ObservationTimeOver wird in dem Zyklus auf FALSE zurückgesetzt, in dem ein Trigger erfasst wird. **Hinweise:** • In dem Zyklus wird auch der Ausgang Position zurückgesetzt, so dass der Signalverlauf von *Position* i. a. einen Sprung aufweist. • Wenn das Signal Position z. B. als Eingangssignal *AxisGroup-Kin.Inst[..].In.Transform.WCS_PCS1 [1]* verwendet wird, darf die Umschaltung in das Koordinatensystem PCS1 erst mehrere Zyklen der TaskPriority später erfolgen, um der Kinematiksteuerung ausreichend Zeit zur Beobachtung der aktuellen Bewegung im neuen Koordinatensystem PCS1 zu geben. • Das Ausgangssignal *ObservationTimeOver* wird mehrere Zyklen nach Erfassung des Triggers auf TRUE gesetzt. So ist die Umschaltung in das neue Koordinatensystem (z. B. PCS1) zulässig, nachdem *Standby* auf TRUE gesetzt wurde, wenn *TriggerOccurred* = TRUE und *ObservationTimeOver* = TRUE sind.

20211732 / DE – 07/2014

12.4 Funktionsbaustein MC_KinWcsPcs1Assignment

Der Funktionsbaustein MC_KinWcsPcs1Assignment weist der Eingangsvariablen *AxisGroupKin.Inst[..].In.Transform* des Programmmoduls AxisGroupControl Kinematics die Transformation aus WCS zu dem im Eingangssignal *TransformIndex* spezifizierten Koordinatensystem zu.

Der Funktionsbausstein bietet sich in Anwendungen an, deren Programmierung der Bewegungsabläufe in mehr als 2 Piece Coordinate Systemen PCS erfolgt (max. KIN_TRANSFORM_ARRAY_SIZE = 10, kann durch gleichnamige Konstante mit anderem Wert verschattet werden.).

Die sprunghafte Änderung des Verlaufs von *AxisGroupKin.Inst[..].In.Transform.WCS_PCS1/2* darf nur erfolgen, während die Steuerung der Kinematik in einem anderen Koordinatensystem erfolgt. Theoretisch kann so die Umschaltung zwischen beliebig vielen, verschiedenen Koordinatensystemen erfolgen. Allerdings erfordert dies die kontinuierlich veränderliche Zuweisung zu PCS1/2. Um diese Veränderlichkeit zu eliminieren, können Sie die Transformationen zu den zahlreichen Koordinatensystemen Ihrer Anwendung in einem Array halten oder zyklisch aktualisieren, wobei jede Transformation ihren festen, von Ihnen definierten Platz innerhalb des Arrays hat. Während des Ablaufs können Sie dann mittels des Funktionsbausteins MC_KinWcsPcs1Assignment oder MC_KinWcsPcs2Assignment dem jeweils aktuell nicht genutzten Koordinatensystem PCS1 oder PCS2 die Transformation zu dem von Ihnen gewünschten Koordinatensystems zuweisen.

HINWEIS

Der Funktionsbaustein MC_KinWcsPcs1Assignment muss zyklisch in der TaskPriority ausgeführt werden.

Das Eingangssignal *TransformIndex* hingegen wird in der Task belegt, in der der Bewegungsablauf programmiert ist. Lassen Sie dazu entweder den Eingang *TransformIndex* leer (Löschen des Fragezeichens nach Einfügen des -Funktionsbausteins) und greifen Sie aus der anderen Task direkt auf den Eingang TransformIndex zu (z. B. fb_WcsPcs1Assignment.CoordSys = COORDSYS_CONVEYORBELT) oder verwenden Sie für die Übergabe eine Variable.

Die aktuellen Transformationen aus WCS zu den Koordinatensystemen werden einem Array zugewiesen, das am Ein-/Ausgangssignal *TransformArray* angelegt wird. Bei stetig veränderlichen Transformationen (z. B. Verwendung des Ausgangs *Position* des Funktionsbausteins MC_KinEncoderDataProcessing) muss diese Zuweisung zyklisch in der TaskPriority erfolgen.

Für größtmöglichen Programmkomfort können Sie für die Indizes des Arrays eine Enumeration definieren, um mittels verständlichen Bezeichnern auf die in Ihrer -Anwendung verwendeten Transformationen zuzugreifen (z. B. COORDSYS_CONVEYORBELT, COORDSYS_MOVINGOBJECT, COORDSYS_RACK, ...)

Sobald eine neue Transformation am Eingang *TransformIndex* ausgewählt wird, wird das Ausgangssignal *ObservationTimeOver* auf FALSE zurückgesetzt. Nachdem die Kinematiksteuerung ausreichend Zeit für die Beobachtung der aktuellen Bewegung im neuen Koordinatensystem hatte (mehrere Zyklen der TaskPriority), wird der Ausgang *ObservationTimeOver* wieder auf TRUE gesetzt.

20211732 / DE – 07/2014

Nachdem das gewünschte Koordinatensystem zugewiesen wird (fb_WcsPcs1Assignment.CoordSys = 'desired coord.sys') und die Beobachtungszeit verstrichen ist (fb_WcsPcs1Asignment.ObservationTimeOver = TRUE), ist die Umschaltung der Kinematiksteuerung in PCS1 zulässig (*AxisGroupKin.Inst[..].In.Target.CoordSys = KIN_PCS1*).

Eingangssignal	Typ	Bedeutung
EN	BOOL	TRUE: Der Funktionsbaustein wird ausgeführt. FALSE: Der Funktionsbaustein wird nicht ausgeführt.
TransformIndex	INT	Index des Koordinatensystems, das PCS1 zugewiesen werden soll. Der Index bezieht sich auf das Array, das im Ein-/Ausgangssignal *TransformArray* übergeben wird.

Ausgangssignal	Typ	Bedeutung
ObservationTimeOver	BOOL	TRUE: Wenn zusätzlich der Ausgang *AssignedTransform* dem geforderten *TransformIndex* entspricht, ist die Umschaltung des Koordinatensystems in PCS1 zulässig FALSE: Umschaltung in PCS1 nicht zulässig.
AssignedTransform	INT	Index des Koordinatensystems, das aktuell der Variable *AxisGroupKin.Inst[..].In.Transform.WCS_PCS1* zugewiesen wird.
Error	BOOL	TRUE: TransformIndex hat einen Wert < 1 oder > KIN_TRANSFORM_ARRAY_SIZE. In diesem Fall weist der Funktionsbaustein der Variablen *AxisGroupKin.Inst[..].In.Transform.WCS_PCS1* keine Werte zu. **Hinweis:** Dieser Fehler verschwindet automatisch, sobald ein gültiger *TransformIndex* übergeben wird.

Ein-/Ausgangssignal	Typ	Bedeutung
Transform	MC_KIN_IN_TRANSFORM	Variable *AxisGroupKin.Inst[..].In.Transform*
TransformArray	ARRAY[1..KIN_TRANSFORM_ARRAY_SIZE] OF ARRAY[1..6] OF LREAL	Array, das die Transformationen von WCS zu den anwendungsspezifischen Koordinatensystemen beinhaltet.

12.5 Funktionsbaustein MC_KinWcsPcs2Assignment

Der Funktionsbaustein MC_KinWcsPcs2Assignment verhält sich exakt identisch zum Funktionsbaustein MC_KinWcsPcs1Assignment. Es erfolgt lediglich die Zuweisung zu WCS_PCS2 statt zu WCS_PCS1.

20211732 / DE – 07/2014

12.6 Funktionsbaustein MC_KinProfGen

Der Funktionsbaustein MC_KinProfGen erzeugt ein ruckbegrenztes Positionsprofil für eine Dimension. Der Profilgenerator verwendet kinematische Positions- und Bewegungswerte in LREAL-Genauigkeit. Er kann mit jeder Startgeschwindigkeit und Startbeschleunigung initialisiert werden.

Eingangssignal	Typ	Bedeutung
EN	BOOL	TRUE: Der Funktionsbaustein wird ausgeführt. FALSE: Der Funktionsbaustein wird nicht ausgeführt.
FeedEnable	BOOL	TRUE*: ActPo*s wird zur *TargetPos* verfahren FALSE: Die Bewegung wird abgebremst mit den Rampen *Dma*x und *JerkTime*.
RapidStop	BOOL	TRUE: Die Bewegung wird abgebremst mit den Rampen *D_Rapid*, *JerkTime_Rapid.* FeedEnable wird intern auf FALSE gesetzt. FALSE: Keine Auswirkung, d. h. *FeedEnable* ist gültig.
TargPos	LREAL	Zielposition in frei wählbarer Anwendereinheit. Die Anwendereinheit wird nicht explizit als „mm“, „m“ oder Ähnliches gewählt. Statt dessen wird durchgängig die gleiche Einheit für alle Signale des Funktionsbausteins verwendet.
Vmax	LREAL	Maximale Geschwindigkeit für das Verfahren zur Zielposition. Einstellung > 0 (Anwendereinheit/Anwenderzeitbasis) Wird die Bewegung mit einer Geschwindigkeit > *Vmax* initialisiert, bremst der Profilgenerator auf *Vmax* ab.
Amax	LREAL	Maximale Beschleunigung (Zunahme der Bewegungsenergie). Einstellung > 0 (Anwendereinheit/Anwenderzeitbasis^2)
Dmax	LREAL	Maximale Bremsbeschleunigung (Abnahme der -Bewegungsenergie). Einstellung > 0 (Anwendereinheit/Anwenderzeitbasis^2)
JerkTime	LREAL	Ruckzeit (Zeit für die Veränderung der Beschleunigung). Einstellung ≥ 0 (Anwenderzeitbasis)
D_Rapid	LREAL	≥ *Dmax*, verwendet statt *Dmax* zur Vermeidung von Überschwingen oder wenn *RapidStop* = TRUE.
JerkTime_Rapid	LREAL	≤ *JerkTime*, verwendet statt *JerkTime* zur Vermeidung von Überschwingen oder wenn *RapidStop* = TRUE.
RapidDynamicsAtOvershooting	BOOL	FALSE: Selbst bei Erkennen von Überschwingen werden *Dmax* / *JerkTime* verwendet. TRUE: Bei Erkennen von Überschwingen werden *D_Rapid* / *JerkTime_Rapid* verwendet.
ModuloMode	MC_KIN_DIRECTION	KIN_SHORT, KIN_CW, KIN_CCW

Ausgangssignal	Typ	Bedeutung
ActPos	LREAL	Aktueller Positionswert des Profilgenerators
ActVel	LREAL	Aktueller Geschwindigkeitswert des Profilgenerators (Anwendereinheit/Anwenderzeitbasis)

20211732 / DE – 07/2014

Ausgangssignal	Typ	Bedeutung
ActAcc	LREAL	Aktueller Beschleunigungswert des Profilgenerators (Anwendereinheit/Anwenderzeitbasis^2)
TargPosReached	BOOL	TRUE: Zielposition erreicht mit *ActPos* = *TargetPos* und *ActVel* = 0 und *ActAcc* = 0. FALSE: Zielposition nicht erreicht.
Overshooting	BOOL	TRUE: Der Profilverlauf *ActPos* entfernt sich momentan von der Zielposition *ActPos*. FALSE: Kein Überschwingen.
ModuloPhaseChange	SINT	0: kein Phasenwechsel −1: negativer Phasenwechsel 1: positiver Phasenwechsel
Error	BOOL	TRUE: Die Bewegung wird abgebremst unter Verwendung von *D_Rapid* / *JerkTime_Rapid* **Hinweis:** Fehler-Reset ist nur möglich durch Neu-Initialisierung.
ErrorID	DWORD	Fehler ID

20211732 / DE – 07/2014

Ausgangssignal	Typ	Bedeutung
Init	MC_KIN_PROFGEN_INIT	Die Initialisierung des Profilgenerators erfolgt durch die Belegung der folgenden Strukturkomponenten: • *New_Init*: BOOL Pegelgesteuerte Initialisierung des Profilgenerators TRUE: Neu-Initialisierung FALSE: Keine Initialisierung **Hinweis:** *New_Init* muss nach der Initialisierung explizit auf FALSE zurückgesetzt werden, ansonsten wird in jedem Zyklus initialisiert. Bereits im Zyklus, in dem die Initialisierung erfolgt (*New_Init* = TRUE), werden die ersten Profilwerte berechnet und in *ActPos* ausgegeben. • *DeltaT*: LREAL Virtueller Zeitfortschritt pro Ausführung des Profilgenerators Einstellung > 0 (Anwenderzeitbasis) Beispiele: Interpolationszeit: 5 ms (Anwenderzeitbasis: s) *DeltaT* = 0.005 Interpolationszeit: 10 ms (Anwenderzeitbasis: ms) *DeltaT* = 10 • *InitPos*: LREAL Initialposition (Anwendereinheit) • *InitVel*: LREAL Initialgeschwindigkeit (Anwendereinheit/Anwenderzeitbasis) • *InitAcc*: LREAL Initialbeschleunigung (Anwendereinheit/Anwenderzeitbasis^2) • *Modulo*: BOOL TRUE: Absolute Modulo-Positionierung Hinweis: (Modulo = TRUE) und (New_Init = TRUE) erfordert: ModuloUnderflow ≤ InitPos ≤ ModuloOverflow False: Absolute Positionierung • *ModuloUnderflow*: LREAL Modulo-Grenzwert für negativen Phasenwechsel • *ModuloOverflow*: LREAL Modulo-Grenzwert für positiven Phasenwechsel

20211732 / DE – 07/2014

13 Anwendungsfälle

13.1 Referenzieren eines Roller-Gantry

Interpolierendes, kartesisches Verfahren ist beispielsweise bei SCARA- oder Knickarm-Kinematiken erst dann korrekt möglich, wenn alle Einzelachsen korrekt referenziert und konfiguriert sind. Nur dann führen die trigonometrischen Berechnungen der Kinematik-Transformationen zu korrekten Ergebnissen, so dass z. B. eine Linearinterpolation auch tatsächlich zum Abfahren einer Geraden im 3D-Raum führt.

Die Roller-Gantry Kinematik nimmt eine Sonderstellung ein, da auch bei nicht korrekt referenzierten Achsen interpolierendes kartesisches Verfahren z. B. entlang den KCS-X-, -Y-, -Z-Achsen möglich ist. Um kartesisch verfahren zu können, müssen die Achsen referenziert sein. Dies kann z. B. durch Referenzieren mittels Typ 5/8 an der Stelle geschehen, an der die Achsen zufällig stehen. Im kartesischen Raum steht der TCP in der Null-Position, wenn die Achsinkremente den Wert 0 aufweisen (wenn *AxisGroupKin.Inst[..].Config.Axis.KinModelOffset[i], Cart.Offset_KCS[i]* gleich 0).

Statt nun die Einzelachsen zu referenzieren, können Sie den TCP des Roller-Gantry kartesisch im Raum zu Referenznocken verfahren. Durch erneutes Referenzieren an der Stelle, an der die Referenznocken schalten, verschieben Sie die kartesische Null-Position des TCP an diese Stelle.

Alternativ können Sie z. B. unter Kenntnis der beim Schalten der Referenznocken erreichten kartesischen Positionen an die Stelle im Arbeitsraum verfahren, in die Sie die Null-Position legen möchten und dort erneut referenzieren. Kann Ihre Maschine die von Ihnen gewünschte Null-Position mechanisch nicht anfahren, können Sie alternativ z. B. auch eine entsprechende Transformation von WCS nach KCS definieren und Ihr Programm im WCS-Koordinatensystem ausführen.

13.2 Offene Parallelkinematik

Wenn sich die Enden der Teilketten einer Parallelkinematik, z. B. die Verbindungsstellen der beiden Arme einer DELTA-Kinematik (siehe DELTA (→ 🗎 64)) aufgrund der Armlängen, Abstände oder Achswerte nicht berühren können, ist es bei korrekter Konfiguration nicht möglich, eine kartesische Position des TCP zu bestimmen. Entsprechend ist in der Situation auch keine kartesische Interpolation (z. B. KIN_JOG/ TARGET_CART, KIN_LIN/CIRC) möglich.

Wenn Sie das in dieser Situation vorliegende Handikap vorübergehend ignorieren, indem Sie die Eingangsvariable *AxisGroupKin.Inst[..].In.General.Ignore.KinHandicap* auf TRUE setzen, können Sie die Achsen trotzdem in Achsen-Interpolation (KIN_JOG/ TARGET_AXIS) verfahren. Sobald die kinematische Kette der Parallelkinematik geschlossen werden kann, sollten Sie die Variable *AxisGroupKin.Inst[..].In.General.Ignore.KinHandicap* unbedingt wieder auf FALSE zurücksetzen.

Wenn eine kartesische Position berechnet werden kann, ist die Ausgangsvariable *AxisGroupKin.Inst[..].Out.General.Position.KCS_WCS_PCS_Valid* = TRUE.

13.3 Wechsel des Koordinatensystems

Die Bewegungsabläufe werden in Bezug auf ein Koordinatensystem programmiert und ausgeführt. Beispielsweise ist als Default-Koordinatensystem KIN_KCS eingestellt, das sich gewöhnlich im Sockel der Kinematik befindet. Ob sich das ausgewählte Koordinatensystem in Ruhe befindet, oder ob es sich relativ zum WCS bewegt, spielt dabei steuerungstechnisch keine Rolle, solange die eingesetzte Kinematik die programmierten Bewegungen ausführen kann.

20211732 / DE – 07/2014

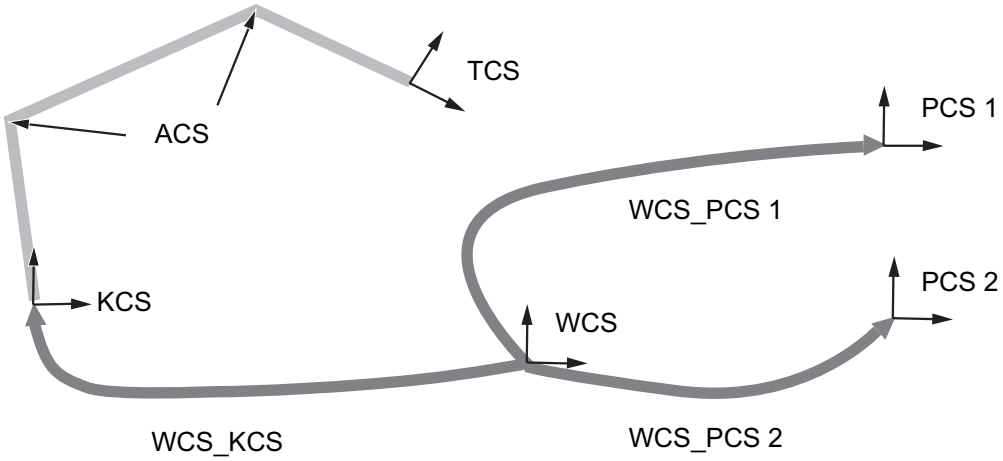
Für die Programmierung stehen insgesamt 6 verschiedene Koordinatensysteme zur Verfügung (siehe folgende Abbildung). Es wird zwischen kinematikbezogenen Koordinatensystemen und umgebungsbezogenen Koordinatensystemen unterschieden.

Kinematikbezogene Koordinatensysteme:

- KIN_KCS: Kinematics Coordinate System (oft im Sockel der Kinematik)
- KIN_ACS: Axis Coordinate System (Achswerte)
- KIN_TCS: Tool Coordinate System (im Werkzeug TCP)

Umgebungsbezogene Koordinatensysteme:

- KIN_WCS: World Coordinate System (gemeinsam für Anlage)
- KIN_PCS1: Piece Coordinate System 1 (Werkstück 1)
- KIN_PCS2: Piece Coordinate System 2 (Werkstück 2)



12549710731

Hinweise:

- Die Programmierung in KIN_ACS bedeutet, dass Zielpositionen oder ein Kreismittelpunkt in Form von Achswerten einzugeben sind. Unabhängig von der Form der Eingabe (Achswerte oder kartesische Koordinaten) erfolgt die Angabe der Interpolation zum Erreichen des Zielpunkts über die Betriebsart.

  So ergibt sich die folgende Kombinatorik der Eingabe von Koordinaten und der Art der Interpolation:

20211732 / DE – 07/2014

Art der Interpolation	Eingabe von Koordinaten	
	CoordSys = KIN_ACS	**Kartesisches Koordinatensystem**
KIN_TARGET_AXIS Bei dieser Interpolation kommt es zu einem Wechsel der KinConstellation, wenn die Achswerte zu Beginn und Ende der Interpolation verschiedenen Konstellationen angehören.	Achsweises Verfahren zu den Ziel-Achswerten	Achsweises Verfahren zu den Ziel-Achswerten, welche der eingegebenen kartesischen Position entsprechen. Für die Umwandlung der kartesischen Position in die Achswerte wird die vom Anwender eingegebene *KinConstellation* und *AxisPhase* verwendet. Die Achsinterpolation erfolgt auf dem kürzesten Weg zu den ermittelten Achswerten.
Kartesische Interpolation Bei dieser Interpolation wird die ursprüngliche KinConstellation beibehalten.	Kartesisches Verfahren (z. B. linear) zur kartesischen Position, welche den eingegebenen Achswerten entspricht – bzw. Umwandlung der eingegebenen Achswerte in einen Kreismittelpunkt.	Kartesisches Verfahren zur kartesischen Zielposition. Die kartesische Interpolation erfolgt komponentenweise (X, Y, Z, A, B, C).

Die Auswahl des Koordinatensystems KIN_TCS ist z. B. nützlich beim kartesischen Tippen KIN_JOG_CART, um den TCP relativ zu bewegen.

Die Transformationen zwischen WCS und KCS/PCS1/PCS2 werden in die Variable *AxisGroupKin.Inst[..].In.Transform.WCS_KCS*, *WCS_PCS1, WCS_PCS2* in Form 6-dimensionaler Vektoren eingespeist. Die Vektoren drücken jeweils die Lage von KCS/PCS1/PCS2 im Koordinatensystem WCS aus. Die ersten 3 Komponenten stellen die Verschiebung des Ursprungs von WCS in X-, -Y-, -Z-Richtung dar. Komponente 4 drückt die Drehung von WCS um die WCS-Z-Achse aus. Das so verdrehte Koordinatensystem wird anschließend mit dem Wert in Komponente 5 um die neue Y-Achse gedreht. Schließlich erfolgt eine Drehung um die neue X-Achse mit dem Wert in Komponente 6.

Die Transformationen WCS_KCS, WCS_PCS1 und WCS_PCS2 können statisch sein oder zur Laufzeit geändert werden, um z. B. das Werkstückkoordinatensystem PCS1 mit einem bewegten Werkstück basierend auf Sensor-Informationen mitzuführen. In dem Fall muss die Veränderung der aktuell genutzten Transformation stetig erfolgen. Sobald die Steuerung in WCS, PCS1 oder PCS2 erfolgt (Ausgang *AxisGroupKin.Inst[..].Out.General.Kin.CoordSys*), versucht die Kinematiksteuerung, die Veränderung der relevanten Transformationen durch entsprechendes Verfahren der beteiligten Achsen auszugleichen. Entsprechend führen sprunghafte Änderungen der relevanten Transformationen zu sprunghaften Bewegungen der Achsen. Relevant sind bei Steuerung in WCS die Transformation WCS_KCS, bei Steuerung in PCS1 die Transformationen WCS_KCS und WCS_PCS1 sowie bei Steuerung in PCS2 die Transformationen WCS_KCS und WCS_PCS2. Die nicht-aktuellen Koordinatensysteme (z. B. WCS_PCS2 bei Steuerung in PCS1) dürfen sprunghaft geändert werden, um z. B. auf ein anderes Werkstück umzuschalten.

20211732 / DE – 07/2014

HINWEIS

Sie müssen einen stetigen Verlauf, der für die Steuerung in einem bestimmten Koordinatensystem relevanten Transformationen bereits mehrere Zyklen sicherstellen, bevor Sie die Steuerung in dieses Koordinatensystem umschalten. Diese Zeit ist erforderlich für die Beobachtung der aktuellen Bewegung im neuen Koordinatensystem, um den Übergang ruckfrei gestalten zu können. Je nach Programmstruktur handelt es sich um 3 bis 5 Zyklen der TaskPriority.

Um Sie bei der Sicherstellung der erforderlichen Beobachtungszeit zu unterstützen, bieten die Funktionsbausteine MC_KinEncoderDataProcessing, MC_KinWcsPcs1Assignment, MC_KinWcsPcs2Assignment das Ausgangssignal *ObservationTimeOver*.

Der Wechsel des Koordinatensystems erfolgt einfach durch Angabe des gewünschten Koordinatensystems am Eingang AxisGroupKin.Inst[..].In.Target.CoordSys bzw. AxisGroupKin.Inst[..].In.Jog.CoordSys, abhängig von der angewählten Betriebsart. Ab diesem Zyklus beziehen sich die programmierten Bewegungen auf das neue Koordinatensystem.

Mit den beiden Werkstück-Koordinatensystemen KIN_PCS1/2 lassen sich Wechsel zwischen den Koordinatensystemen beliebig vieler Werkstücke realisieren, indem das jeweils nicht verwendete Koordinatensystem auf ein anderes Werkstück umgeschaltet wird.

Neben den Transformationen WCS_KCS/PCS1/PCS2 ist die Werkzeugtransformation *AxisGroupKin.Inst[..].In.Transform.Tool* Teil der kinematischen Kette. Sie stellt die Transformation zwischen dem Flansch-Koordinatensystem FCS und dem Tool-Koordinatensystem TCS dar. Bei kartesischer Interpolation (KIN_JOG/TARGET_CART, KIN_LIN_xx, KIN_CIRC_xx) darf die Werkzeugtransformation nur stetig verändert werden, da ansonsten sprunghafte Ausgleichbewegungen der Achse erfolgen. Bei Achsinterpolation (KIN_JOG/TARGET_AXIS, KIN_AXIS_CP) oder in KIN_AM_DEFAULT darf die Werkzeugtransformation sprunghaft verändert werden. In dem Fall springt auch die kartesische Position des Roboters (*AxisGroupKin.Inst[..].Out.General.Position → alle außer ACS/ACS_Incr*).

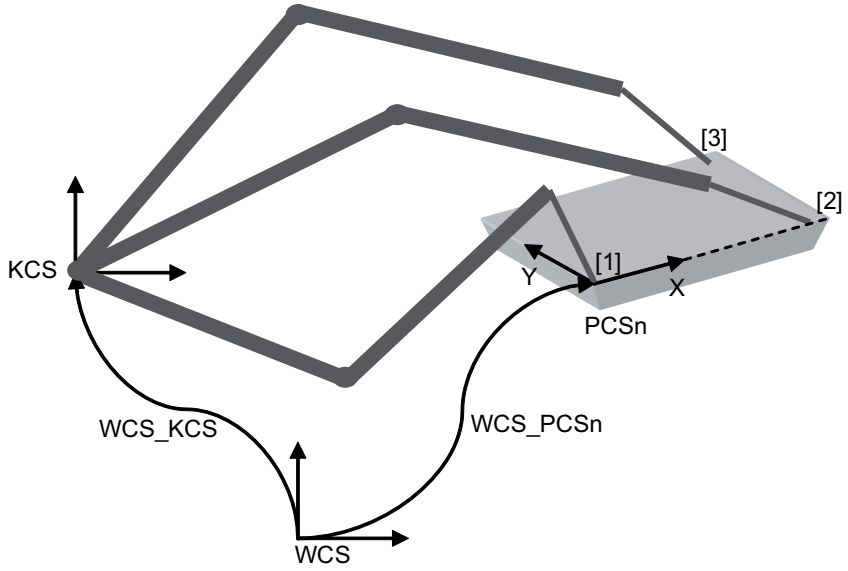
13.4 Vermessen eines Koordinatensystems

Die Vermessung eines Koordinatensystems kann z. B. dazu genutzt werden, abzufahrende Bahnen nur einmal in diesem Koordinatensystem zu programmieren. Das Koordinatensystem kann z. B. an einem Eck einer Palette liegen. Wird die Palette verschoben, muss lediglich ihre Lage in Bezug auf WCS neu vermessen werden. Das programmierte Packmuster kann jedoch weiterhin unverändert verwendet werden.

Zur Vermessung der Lage eines Koordinatensystems in Bezug auf ein anderes Koordinatensystem (z. B. PCS1 in WCS, also zur Bestimmung der Transformation WCS_PCS1) steht der Funktionsbaustein MC_KinCoordSysMeasurement zur Verfügung. Der Funktionsbaustein benötigt 3 Positionen aus Sicht des Koordinatensystems, auf welches sich die Transformation zu dem zu vermessenden Koordinatensystem bezieht. Bei den 3 Positionen handelt es sich um den Ursprung, einen Punkt auf der positiven X-Achse sowie einen Punkt in der XY-Ebene mit positiver Y-Koordinate.

20211732 / DE – 07/2014

Zur Bestimmung dieser Positionen können Sie z. B. den TCP der Kinematik im Tippbetrieb zu diesen Positionen verfahren und die jeweils erreichte Istposition im Koordinatensystem auslesen, auf das sich die Transformation beziehen soll.



12549714827

[1] Ursprung des Koordinatensystems PCS1/2
[2] Punkt auf positiver X-Achse des Koordinatensystems PCS1/2
[3] Punkt in XY-Ebene des Koordinatensystems PCS1/2 mit positiver Y-Koordinate

Hinweise:

Wenn z. B. PCS1/2 in Bezug auf WCS mit dem TCP der Kinematik vermessen werden soll, muss die Transformation WCS_KCS bereits korrekt ermittelt und in die Variable *AxisGroupKin.Inst[..].In.Transform.WCS_KCS* eingespeist sein.

Die höchste Genauigkeit erreichen Sie, wenn der Punkt auf der positiven X-Achse möglichst weit vom Ursprung und der Punkt in der XY-Ebene möglichst weit von der X-Achse entfernt ist. Wenn Sie für die Vermessung den TCP der Kinematik verwenden, sollten die Messpunkte jedoch nicht deutlich außerhalb des Arbeitsraums liegen, in dem das Handling oder die Bearbeitung im zu vermessenden Koordinatensystem erfolgt, um Ungenauigkeiten bei der Modellierung der Kinematik (z. B. Armlängen, Durchbiegungen) zu reduzieren.

Die Orientierung, in welcher der TCP der Kinematik die Messpunkte anfährt, hat theoretisch keine Bedeutung, da nur die X-, Y-, Z-Koordinaten für die Berechnung der Transformation verwendet werden. Allerdings sollte die Orientierung nicht deutlich vom Bereich der Orientierungen abweichen, in denen das Handling oder die Bearbeitung erfolgt, um auch hier Ungenauigkeiten bei der Modellierung der Kinematik und des Werkzeugs zu reduzieren.

Weitere Informationen entnehmen Sie dem Kapitel „Funktionsbaustein MC_KinCoordSysMeasurement (→ 🗎 117).

20211732 / DE – 07/2014

13.5 Programmstruktur für Tracking-Anwendungen mit PCS1/2

Tracking bezeichnet die Ausführung von Handling- oder Bearbeitungsaufgaben bei Synchronisation auf bewegte Werkstücke.

Die Synchronisation erfolgt durch Steuerung im Werkstückkoordinatensystem PCS1/2, das mit dem Werkstück bewegt wird. Hierzu müssen Sie die Transformation WCS_PCS1/2 zyklisch in die Variable *AxisGroupKin.Inst[..].In.Transform.WCS_PCS1/2* einspeisen. Bei bekannter Bewegung des Werkstücks (z. B. konstante Geschwindigkeit entlang WCS-X-Richtung) können Sie die entsprechenden Komponenten des Vektors WCS_PCS1/2 z. B. zyklisch in der TaskPriority inkrementieren. Variiert die Bewegung des Werkstücks, muss sie eventuell über Sensoren zyklisch erfasst werden. Hierbei sind verrauschte Sensorsignale in geeigneter Weise zu glätten, bevor sie in den Vektor WCS_PCS1/2 geschrieben werden.

Zur Aufbereitung von Gebersignalen steht der Funktionsbaustein MC_KinEncoderDataProcessing zur Verfügung. Der Funktionsbaustein muss zyklisch in der TaskPriority ausgeführt werden und wandelt die Geberwerte in die zurückgelegte Strecke in den von Ihnen ausgewählten Anwendereinheiten um.

Die folgende Abbildung zeigt die Aufteilung des Programms in die zyklische Zuweisung der Transformationen in der Interpolations-Task und das Umschalten in das jeweils gewünschte Koordinatensystem in der TaskMain.

TaskPriority

```
…

(* Execution of FB MC_KinEncoderDataProcessing,
therefore e.g. cyclic reading of SCOM *)

fbMC_Receive_CAN(…);
fbMC_KinEncoderDataProcessing(…);

(* Assignment WCS_PCS1 *)
AxisGroupKin.Inst[..].In.Transform.WCS_PCS1[1]
  := fbKinEncoderDataProcessing.Position;

…


(* Assignment WCS_PCS2 *)
AxisGroupKin.Inst[..].In.Transform.WCS_PCS2[1] := 500.0;

…
```

TaskMain

```
…

CASE (MotionSequenceState) OF…

10: … (* Motion commands *) …

20: (* Switching to PCS1 in a TARGET mode,
       IF OBSERVATION TIME OVER *)
    AxisGroupKin.Inst[..].In.Target.CoordSys := KIN_PCS1;

30: … (* Motion commands *) …

40: (* Switching to PCS2 in a TARGET mode,
       IF OBSERVATION TIME OVER *)
    AxisGroupKin.Inst[..].In.Target.CoordSys := PCS2;

50: … (* Motion commands *) …

END_CASE
```

12549766155

In dem Beispiel wird die X-Komponente der Transformation WCS_PCS1 zyklisch aus Geberwerten aktualisiert. Die X-Komponente der Transformation WCS_PCS2 hat hingegen einen festen Wert. Entsprechend sind alle anderen Komponenten der Transformationen zuzuweisen. Während der Bewegungsablauf in PCS2 erfolgt, können Sie die Transformation WCS_PCS1 z. B. auf ein anderes bewegtes Werkstück umschalten. Um z. B. zwischen beliebig vielen verschiedenen, bewegten Werkstücken umzu-

20211732 / DE – 07/2014

schalten (z. B. Entnahme von Backwaren von einem Band), können Sie auch das Koordinatensystem PCS2 mit dem Werkstück mitführen. Die sprunghafte Zuweisung einer neuen Transformation zu WCS_PCS1/2 (für ein anderes Werkstück) ist immer dann zulässig, während die Steuerung in einem anderen Koordinatensystem erfolgt.

Hinweise:

Beachten Sie bei der Implementierung bitte auch die im Kapitel "Wechsel des Koordinatensystems“ erläuterten Beobachtungszeiten, die für ruckfreies Umschalten einzuhalten sind.

Der Wechsel des Koordinatensystems ist in einer ContinuousPath Betriebsart nicht möglich. Die Umschaltung erfolgt im Automatikbetrieb mittels KIN_TARGET_CART oder KIN_TARGET_AXIS. Sobald die Bewegung im neuen Koordinatensystem für einen Moment zum Stillstand gekommen ist (*AxisGroupKin.Inst[..].Out.General.Standstill.ActCoordSys* = TRUE), können Sie in ContinuousPath umschalten. Dies bedeutet nicht, dass die Achsen oder der TCP relativ zu KCS stehen müssen. Lediglich die Bewegung in einem bewegten Koordinatensystem muss für einen Moment zum Stillstand kommen.

Weitere Informationen entnehmen Sie bitte dem Kapitel "Funktionsbaustein MC_KinEncoderDataProcessing".

13.6 Programmstruktur für Tracking-Anwendungen mit mehr als 2 PCS

Zur komfortablen Verwaltung von mehr als 2 Werkstückkoordinatensystemen stehen die Funktionsbausteine MC_KinWcsPcs1/2Assignment zur Verfügung.

Wenn in Ihrer Anwendung z. B. Werkstückkoordinatensysteme erforderlich sind für

- Ablage auf Band 1
- Ablage auf Band 2
- Aufnahme von Band 1
- Aufnahme von Band 2
- Palettenplatz 1
- Palettenplatz 2
- usw.,

dann können Sie die entsprechenden Transformationen immer im Wechsel den Eingängen WCS_PCS1 und WCS_PCS2 zuweisen. Zur Erleichterung dieser Verwaltung können Sie alle benötigten Transformationen alternativ auch in einem zentralen Array eintragen oder bei bewegten Objekten zyklisch aktualisieren:

```
TransformArray : ARRAY [1..KIN_TRANSFORM_ARRAY_SIZE] OF ARRAY
[1..6] OF LREAL;

TransformArray[LAYDOWN_CONVEYOR_1] := ...;

TransformArray[LAYDOWN_CONVEYOR_2] := ...;

TransformArray[PICKING_CONVEYOR_1] := ...;

TransformArray[PICKING_CONVEYOR_2] := ...;

TransformArray[PALLET_1] := ...;

TransformArray[PALLET_2] := ...;

usw.
```

Um aussagekräftige Indizes zu nutzen, sollten Sie eine Enumeration Ihrer Koordinatensysteme erstellen.

```
TYPE MY_COORDSYS : (
```

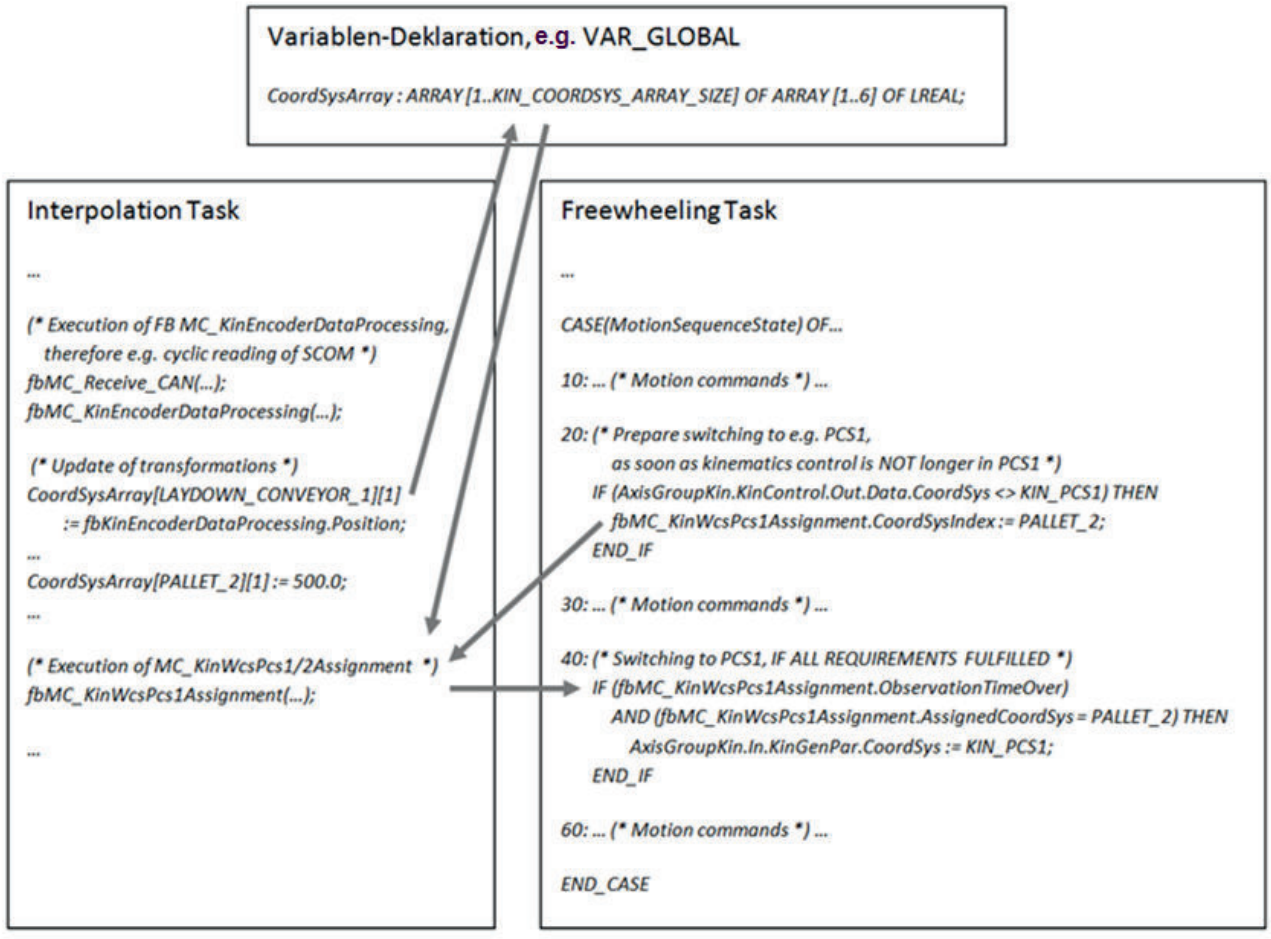
20211732 / DE – 07/2014

```
LAYDOWN_CONVEYOR_1,
LAYDOWN_CONVEYOR_2,
PICKING_CONVEYOR_1,
PICKING_CONVEYOR_2,
PALLET_1,
PALLET_2,
(* usw *));.
```

Die Funktionsbausteine MC_KinWcsPcs1/2Assignment dienen dazu, die jeweils gewünschte Transformation aus dem *TransformArray* den Eingängen WCS_PCS1 und WCS_PCS2 zuzuweisen. Hierzu müssen Sie den Funktionsbausteinen lediglich den Index (Eingang *TransformIndex*) des jeweils gewünschten Koordinatensystems übergeben (z. B. LAYDOWN_CONVEYOR_1).

Für den Handshake zwischen dem Ablaufprogramm in der TaskMain und der Ausführung der Funktionsbausteine MC_KinWcsPcs1/2Assignment in der TaskPriority stehen entsprechende Ausgangssignale zur Verfügung. Die Funktionsbausteine geben aus, welche Transformation aktuell zugewiesen wird (*AssignedTransform*) und ob die seit dem Wechsel des TransformIndex für eine ruckfreie Umschaltung erforderliche Beobachtungszeit verstrichen ist (*ObservationTimeOver*). Sobald die gewünschte Transformation zugewiesen wird und die Beobachtungszeit verstrichen ist, darf die Umschaltung des Koordinatensystems im Bewegungsablaufprogramm in der TaskMain erfolgen.

Die folgende Darstellung zeigt die Programmstruktur für Tracking-Anwendungen mit mehr als 2 PCS:



20211732 / DE – 07/2014

12549853579

Weitere Informationen entnehmen Sie bitte den Kapiteln "Funktionsbaustein MC_KinEncoderDataProcessing“ sowie "Funktionsbaustein MC_KinWcsPcs1/2-Assignment“.

13.7 Verwendung einer Tool-Transformation

Die spezifizierten Kinematikmodelle (→ 🗎 57) umfassen die Transformation vom Kinematik-Koordinatensystem KCS zum Flansch der Kinematik (FCS). Am Flansch wird ein Greif- oder Bearbeitungswerkzeug angebracht. Bei einigen Modellen lässt sich ein einfaches Werkzeug (z. B. Versatz von FCS bis zum Saugnapf eines Vakuumgreifers rein entlang der FCS-Z-Koordinate) bereits in den Kinematik-Parametern abbilden.

Bei komplexeren Werkzeugen müssen Sie der Kinematik-Steuerung explizit die Werkzeug- bzw. Tool-Transformation vom Flansch-Koordinatensystem FCS zum Tool-Koordinatensystem TCS übergeben. Der Tool-Center-Point (TCP) befindet sich im Ursprung des TCS. Programmierte Bahnen werden durch den TCP abgefahren. Die Transformation wird in Form eines Vektors ausgedrückt und beschreibt die relative Lage des TCS in Bezug auf FCS. Die ersten 3 Komponenten drücken die Verschiebung vom Flansch-Center-Point (FCP) zum TCP aus. Die vierte Komponente entspricht der Drehung des FCS um seine Z-Achse, die fünfte Komponente die Drehung um die neue Y-Achse und die sechste Komponente die Drehung um die neue X-Achse. Den Vektor müssen Sie der Eingangsvariable *AxisGroupKin.Inst[..].In.Transform.Tool* zuweisen.

Bei veränderlicher Tool-Transformation darf der Signalverlauf des Transformations-Vektors während kartesischer Interpolation und bereits 5 Zyklen (der TaskPriority) vor kartesischer Interpolation keine Sprünge aufweisen, da die Kinematik-Steuerung versucht, diese Sprünge auszugleichen. Bei Verwendung verrauschter Sensorsignale für die Veränderung der Tool-Transformation müssen Sie diese vor der Einspeisung in die Kinematik-Steuerung in geeigneter Weise glätten.

Hinweis:

Bei einem Werkzeugwechsel muss die Tool-Transformation auf das neue Werkzeug umgeschaltet werden, während sich die Kinematik-Steuerung

- entweder nicht in Interpolation befindet

  oder in Achsen-Interpolation.

Im ersten Fall wird die neue Transformation beim erneuten Wechsel in Interpolation übernommen. Im zweiten Fall wird die neue Transformation direkt übernommen, so dass sich die kartesische Position im gleichen Zyklus schlagartig ändert. Da in dem Moment jedoch die Achsen interpoliert werden, kommt es zu keinem Schleppfehler.

Sie können eine zur Laufzeit veränderliche Tool-Transformation z. B. dazu verwenden, die durch einen im Werkzeug integrierten Antrieb verursachte Änderung der Transformation (z. B. Linearachse zur Veränderung der Länge) zu kompensieren.

Ein anderes Anwendungsszenario für eine veränderliche Tool-Transformation ist die kontinuierliche Ausrichtung des Werkzeugs entlang der Tangente einer Bahn. Hierzu können Sie im Anwenderprogramm die Tangente zyklisch aus den aktuellen Positionen ermitteln und die Tool-Transformation in gewünschter Weise zyklisch anpassen. Applikative Sonderbehandlungen sind dabei evtl. erforderlich, wenn die Bahn beispielsweise Knicke aufweist.

20211732 / DE – 07/2014

13.8 Kombination von Kurvenscheiben- und Kinematikfunktionalität

Wenn mittels tabellarischer oder online gerechneter Kurvenscheiben Konturen abgefahren werden, kommen gewöhnlich kartesische Gantry-Kinematiken zum Einsatz. Dabei werden für jede der rechtwinklig zueinander angeordneten Achsen Kurvenverläufe bestimmt und synchronisiert abgefahren.

Sie können die Kurvenscheiben-Funktionalität mit der Kinematik-Steuerung kombinieren, indem Sie die Positionssollwerte, welche gewöhnlich direkt an die Achsen gesendet werden, als kartesische X-, Y-, Z-Raumkoordinaten in die Kinematik-Steuerung einspeisen. Die Kinematiksteuerung führt wie im „normalen“ Kinematik-Betrieb die inverse Kinematiktransformation zur Bestimmung der zugehörigen Achsverläufe durch und steuert entsprechend die Achsen einer beliebigen Kinematik. Als Ergebnis fährt der TCP der Kinematik die gewünschte Kontur ab.

Hierzu müssen Sie bei angewählter Betriebsart *AxisGroupKin.Inst[..].In.Mode = KIN_TARGET_CART* die Positionssollwerte im Interpolationstakt, also in der TaskPriority, in die Eingangsvariable *AxisGroupKin.In.MasterPosition.Target* kopieren. Durch die Einstellung *AxisGroupKin.Inst[..].In.Target.UseMasterPosition* werden die Achsen in jedem Zyklus in die Stellung bewegt, welche erforderlich ist zum Anfahren der anliegenden kartesischen Sollposition. Wie bei herkömmlichen Kurvenscheiben-Anwendungen müssen die Kurvenverläufe stetig sein.

Hinweis:

Es besteht jederzeit die Möglichkeit, die Variable *AxisGroupKin.Inst[..].In. Target.UseMasterPosition* auf FALSE zurückzusetzen, so dass die Kinematiksteuerung fließend und ruckfrei die Bahnsteuerung übernimmt und die Kinematik interpolierend zu den kartesischen Zielkoordinaten verfährt.

Sie können auch jederzeit z. B. in die Betriebsart KIN_TARGET_AXIS umschalten, um z. B. mit der Einstellung KIN_SYNC_XYZABC_OR_A1TOA6 fließend und ruckfrei in Achsen-Interpolation zu wechseln.

Im Gegensatz zur herkömmlichen Kurvenscheiben-Funktionalität können Sie bei Kombination mit der Kinematik-Steuerung auf einfache Art mit komplexen Kinematiken Konturen auf Objekten abfahren, die sich in Bewegung befinden. Hierzu wechseln Sie z. B. in das Koordinatensystem KIN_PCS1/2. Auch die Kinematik selbst kann dabei von externen Achsen verfahren werden (veränderliche Transformation WCS_KCS). Somit kann es sogar bei Einsatz eines kartesischen Gantry sinnvoll sein, die Kurvenscheiben-Funktionalität mit der Kinematik-Steuerung zu kombinieren.

13.9 CP-Interpolation als Slave eines externen Masterprofils

Bei ContinuousPath Interpolation steuert gewöhnlich die Kinematik-Steuerung den Bahnfortschritt. Es besteht auch die Möglichkeit, die CP-Bahn wie gewohnt einzugeben, jedoch den Bahnfortschritt extern, z. B. mittels eines virtuellen Gebers, einer Masterachse oder eines Sensorsignals zu steuern.

Hierzu muss die Eingangsvariable *AxisGroupKin.Inst[..].In.Cp.Settings.UseMasterPosition* beim Wechsel in eine ContinuousPath Betriebsart gesetzt sein. Das Positionsprofil speisen Sie zyklisch in der Interpolation-Task in die Variable *AxisGroupKin.Inst[..].In.MasterPosition.Cp* ein. Das Positionsprofil muss stetig sein und darf nicht absteigen.

20211732 / DE – 07/2014

13.10 Kontinuierliche Veränderung der Kinematik-Parameter

Sie haben die Möglichkeit, die Kinematik-Parameter zur Laufzeit zu verändern. Hierzu müssen Sie die Variable *AxisGroupKin.Inst[..].Config.CyclicTakeover.Kin_Par* auf TRUE setzen. Bei diesem Pegel wird das Array *AxisGroupKin.Inst[..].Config.Kin.Par* übernommen. Das zyklische Ändern von Parametern kann z. B. in folgenden Situationen nützlich sein:

- Anpassung von Armlängen infolge Temperaturschwankung
- Anpassung von Kinematik-Parametern (z. B. Winkel) infolge einer hohen Last am TCP
- Anpassung von Parametern von Kinematiken mit geringer Steifigkeit in Abhängigkeit der Stellung
- Anpassung von Armlängen wegen eines integrierten Antriebs, der nicht zu den Kinematikachsen zählt

13.11 Kinematiken mit mehr als 6 Antrieben

Die Kinematik-Modelle können maximal 6 verschiedene Antriebe umfassen. Allerdings können Sie auch Kinematiken mit mehr als 6 Achsen steuern. Folgende Möglichkeiten stehen hierzu zur Verfügung:

- Wenn mehrere Antriebe erforderlich sind zur Bewegung einer Achse (z. B. große Momente oder Hallenkräne), können Sie z.B. eine Kinematik-Achse als virtuelle Achse konfigurieren und mehrere reale MultiMotion-Achsen in der MultiMotion-Betriebsart Tracking synchronisiert mit der virtuellen Achse verfahren.
- Die Positionsverläufe von Achsen, welche die komplette Kinematik bewegen, können Sie in die Transformation WCS_KCS einspeisen. So lassen sich z. B. in WCS oder PCS1/2 programmierte Bahnen abfahren. Bei redundanten Achsen können Sie beispielsweise die externe Achse zum Verfahren der kompletten Kinematik abhängig vom Bahnfortschritt z. B. in WCS steuern. Die tatsächlich erreichten Achswerte werden dann wiederum in WCS_KCS rückgespeist, so dass die Bahn in WCS exakt abgefahren wird.
- Zyklische Anpassung der Tool-Transformation für ein Werkzeug mit integriertem Antrieb
- Zyklische Anpassung aller Parameter einer Kinematik (z. B. durch Antriebe bewegte Teleskoparme)

20211732 / DE – 07/2014

13.12 Einbinden einer User-Kinematik

Falls keines der im Kapitel Kinematikmodelle (→ 🗎 57) aufgeführten Modelle Ihrer Kinematik entspricht, können Sie eine User-Kinematik integrieren. Hierzu wird die Bibliothek `MPLCKinematics_UserKin` ausgetauscht. In diesem Fall bitten wir Sie darum, SEW-EURODRIVE zur Beratung zu kontaktieren.

13.13 Umgang mit Mehrdeutigkeiten der ABC-Orientierungswerte

Die Darstellung einer Werkzeugorientierung mittels ABC-Orientierungswerten ist mehrdeutig.

Beispiele:

- (A = 90°, B = 0°, C = 0°) entspricht der gleichen Orientierung des Werkzeugs im Raum wie (A = 450°, B = 0°, C = 0°). Soll ausgehend von dieser Orientierung zu den Zielwerten (A = 150°, B = 0°, C = 0°) komponentenweise interpoliert werden, ergibt sich je nach Start-ABC-Werten eine andere Bewegung.
- (A = 90°, B = 90°, C = 90°) beschreibt dieselbe Orientierung wie (A = 0°, B = 90°, C = 0°). Soll ausgehend von dieser Orientierung zu den Zielwerten (A = 0°, B = 100°, C = 0°) komponentenweise interpoliert werden, ergibt sich je nach Start-ABC-Werten eine andere Bewegung.

ACHTUNG

Mehrdeutigkeiten

Wenn die Mehrdeutigkeiten nicht in Betracht gezogen werden, kann es zu Drehungen in nicht erwarteter Richtung und somit z. B. zum Aufwickeln und Abreißen von Kabeln oder Schläuchen kommen.

Sie haben die folgenden Möglichkeiten, mit diesen Mehrdeutigkeiten umzugehen:

13.13.1 Eindeutige Zuordnung

Konfiguration der eineindeutigen Zuordnung kartesischer Orientierungs-Koordinaten und Achswerten

AxisGroupKin.Inst[..].Config.General.AxisCartBijection = TRUE

AxisGroupKin.Inst[..].Config.General.MapABCToStartUpValuesAtConfig = FALSE

Das ist die Default-Einstellung bei den zahlreichen Kinematik-Modellen.

Vorteile:

- Keine Probleme mit Mehrdeutigkeiten
- Keine Beschränkung der Orientierungs-Winkel

Nachteile:

- Nicht für alle Kinematik-Modelle möglich
- Nicht alle Freiheitsgrade der Transformationen nutzbar
- Wenn Achsen-SWLS verlassen, dann nur Achsen-Interpolation möglich

20211732 / DE – 07/2014

13.13.2 Keine Eindeutigkeit

Keine Eineindeutigkeit und kein Interpolations-Start mit abgespeicherten Orientierungswerten

AxisGroupKin.Inst[..].Config.General.AxisCartBijection = FALSE

AxisGroupKin.Inst[..].Config.General.MapABCToStartUpValuesAtConfig = FALSE

Das ist die Default-Einstellung bei den Kinematik-Modellen, welche Eineindeutigkeit nicht unterstützen.

Vorteile

- Mit allen Kinematimodellen möglich
- Alle Freiheitsgrade der Transformationen nutzbar

Nachteile

- Wenn die Achsen vor Start der Maschine und Wechsel in eine kinematische Betriebsart eventuell verdreht wurden, ist eine achsweise Grundstellungsfahrt (KIN_TARGET_AXIS, Target.CoordSys = KIN_ACS) erforderlich mit anschließendem Wechsel in z. B. AM_DEFAULT (*Out.General.Kin.Ready = FALSE*) und erneutem Wechsel in eine kinematische Betriebsart.
- Wenn kartesische Orientierungswerte ausserhalb des Bereichs]-180°, 180°[angefahren werden sollen, ist die zuvor beschriebene Grundstellungs-Prozedur nach jedem Wechsel in eine kinematische Betriebsart erforderlich.

13.13.3 Übernahme der abgespeicherten Orientierungswerte

Keine Eineindeutigkeit, jedoch Übernahme der abgespeicherten Orientierungswerte bei Interpolations-Start

AxisGroupKin.Inst[..].Config.General.AxisCartBijection = FALSE

AxisGroupKin.Inst[..].Config.General.MapABCToStartUpValuesAtConfig = TRUE

Das ist keine Default-Einstellung.

Vorteile:

- Mit allen Kinematikmodellen möglich
- Alle Freiheitsgrade der Transformationen nutzbar
- Keine Grundstellungsfahrt erforderlich, falls Achsen mit Sicherheit seit Verlassen einer Kinematik-Betriebsart (*Out.General.Kin.Ready* = FALSE) nicht verdreht wurden, kein Tausch der Motorgeber und kein Tausch der MOVI-PLC® erfolgte.

Nachteile:

- Wenn die Achsen vor Start der Maschine und Wechsel in eine kinematische Betriebsart eventuell verdreht wurden oder ein Motorgeber oder die MOVI-PLC getauscht wurde, ist eine achsweise Grundstellungsfahrt (KIN_TARGET_AXIS, Target.CoordSys = KIN_ACS) erforderlich. Anschließend muss die gewünschte Anfangsorientierung mittels der im Folgenden beschriebenen Prozedur explizit eingestellt werden.

Prozedur zum Einstellen einer gewünschten Anfangsorientierung:

Beim Wechsel in die Betriebsart KIN_TARGET_CART (bzw. ContinuousPath mit aktiviertem *Cp.Settings.PlusTargetCartABC*) oder bei Wechsel in ein anderes kartesisches Koordinatensystem in der Betriebsart KIN_TARGET_CART werden die aktuellen Orientierungs-ABC-Werte in die Nähe der Ziel-ABC-Werte im ersten Zyklus gemappt. Um dieses Mapping zu aktivieren, muss *AxisGroupKin.Inst[..].In.Target.ABC-Mapping* = TRUE sein. Damit das Mapping zuverlässig funktioniert, sollte die erste Zielorientierung komponentenweise weniger als 45º von der aktuellen Orientierung in

20211732 / DE – 07/2014

der gewünschten ABC-Darstellung entfernt sein. Sie können z. B. als Ziel im ersten Zyklus die aktuelle Orientierung in der gewünschten ABC-Darstellung angeben und bereits im nächsten Zyklus auf eine hiervon deutlich abweichende Orientierung umschalten.

Sie können die Konfigurationseinstellungen

- Config.General.AxisCartBijection
- Config.General.MapABCToStartUpValuesAtConfig

auf der Expertenseite der Konfiguration an Ihre Anforderungen anpassen (Kapitel „Standardbelegung von Eingangsvariablen (→ 🗎 39)“).

13.14 Sicherung von Resolver-Positionswerten

Zur Sicherung der Resolver-Positionswerte in der MOVI-PLC steht im Technologiemodul Kinematics der Funktionsbaustein AxisGroupKin_WriteSysPosUserUnitsToDDB_MX zur Verfügung. Der Funktionsbaustein kann z. B. in der TaskMain ausgeführt werden. Als Gegenstück können Sie den Funktionsbaustein AxisGroupKin_HomingWithResolver_MX verwenden, welcher in Ihrem Ablaufprogramm in der Hochlaufsequenz dazu dient, die Achsen mit den gespeicherten Werten zu referenzieren.

HINWEIS

- Bei Abschalten der Spannungsversorgung von MOVI-PLC® oder Umrichter muss die Bremse geschlossen sein. Verrutscht die Achse bei geschlossener Bremse oder ändert sich die Achsposition z. B. wegen eines Umbaus, wird bei Verwendung der gespeicherten Positionswerte nicht korrekt referenziert.
- Der Umrichter muss auf Referenzfahrttyp 5/8 (Keine Referenzfahrt) eingestellt sein.
- Die Funktionalität ist in der aktuellen Version nur für MOVIAXIS verfügbar.

20211732 / DE – 07/2014

14 Problembehandlung

14.1 3D-Simulation

14.1.1 Kein Verbindungsaufbau möglich

Problem

Sie können keine Verbindung aufbauen. Im Simulationsfenster links unten wird eine rot markierte „0“ angezeigt.

Abhilfe

Führen Sie nacheinander die folgenden Schritte aus, bis das Problem behoben ist:

Kommunikationseinstellungen überprüfen:

- Stellen Sie eine Ethernet-Verbindung zwischen der MOVI-PLC® und dem Simulations-PC her.

  Hinweis: Die 3D-Simulation ist über USB oder Feldbus nicht möglich!

  Die Kommunikation mit der 3D-Simulation geschieht ausschließlich über die Ethernet-Engineering-Schnittstelle (X37 bei MOVI-PLC® advanced, LAN 3 bei MOVI-PLC® power).

- Stellen Sie die IP-Adresse der Engineering-Schnittstelle des Simulations-PCs in der Konfiguration des Technologiemoduls Kinematics korrekt ein.

  Hinweis: Die eingestellte IP-Adresse des verwendeten Netzwerkadapters können Sie im MotionStudio auslesen:

  – Öffnen Sie den Menübefehl [Netzwerk] / [Kommunikationsanschlüsse konfigurieren].

  – Wählen Sie den Eintrag „Ethernet“ und betätigen Sie die Schaltfläche [Bearbeiten] und anschließend [Netzwerkadapter].

    Es öffnet sich ein Fenster mit den Eigenschaften (IP-Adresse etc.) der verfügbaren Netzwerkadapter.

Firewalleinstellungen überprüfen:

Ein weitere Ursache kann auch die Einstellung Ihrer Firewall sein.

Möglicherweise blockt die Firewall die Kommunikation zwischen MOVI-PLC® und Simulations-PC.

- Deaktivieren Sie die Firewall komplett.

  Oder erstellen Sie mithilfe der Systemsteuerung Ihres Window-Betriebssystems eine Ausnahme für das 3D-Simulations-Programm „SEWDx9“:

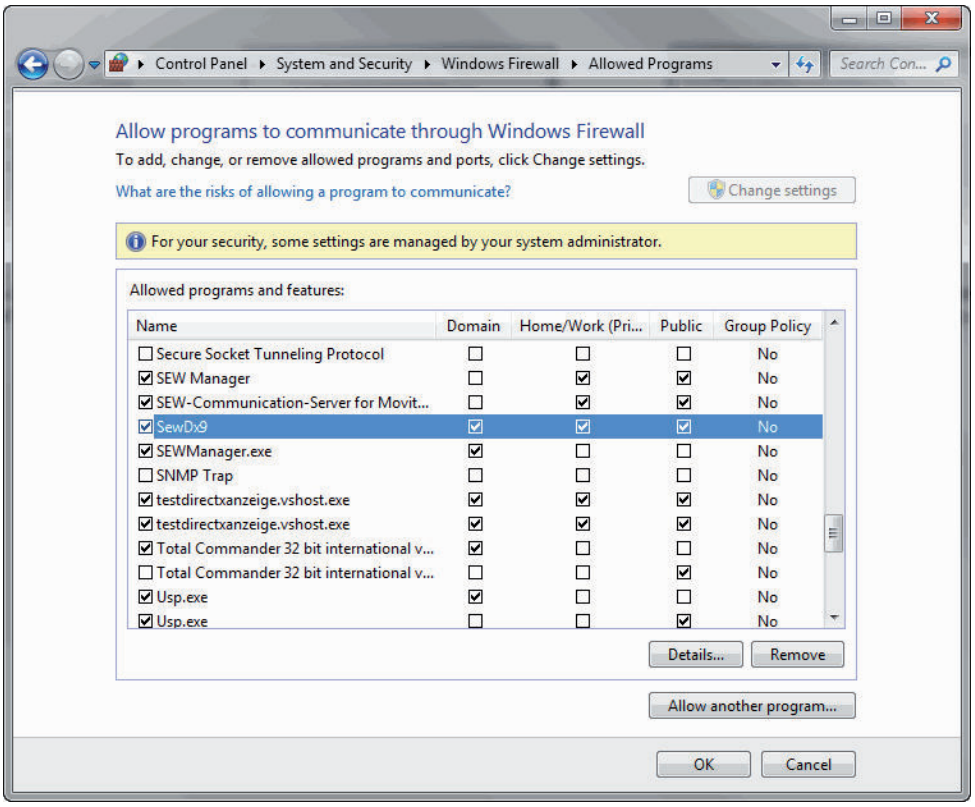
  – Wählen Sie den Eintrag „SEWDx9“ aus der Liste.

  – Sollte der Eintrag nicht enthalten sein, wählen Sie die Exe-Datei aus folgendem Pfad aus:

20211732 / DE – 07/2014

```
C:\Program Files (x86)\SEW\MotionStudio\SewDx9.exe
```

- Aktivieren Sie die Kontrollfelder „Domäne“, „Heim/Arbeit“ und „Öffentlich“:



9308487307

Hinweis: Wenn der Eintrag „SewDx9“ in der Liste mehrfach erscheint, sollten die Kontrollfelder bei allen Einträgen aktiviert sein.

Technologiestufe nicht ausreichend

- Stellen Sie sicher, dass die auf der Speicherkarte hinterlegte Technologiestufe hoch genug ist. Mit weniger als 12 Technologiepunkten kann die 3D-Simulation auf keinen Fall ausgeführt werden.

  Die für die konfigurierte Funktionalität erforderliche Technologiestufe entnehmen Sie entweder dem Kapitel „Erforderliche Technologiestufe (→ 🗎 13) oder der Anzeige im Konfigurationsassistenten.

14.1.2 Es erscheint kein Modell

Problem

Sie können zwar eine Verbindung aufbauen, (im Simulationsfenster links unten wird eine grün markierte Zahl angezeigt), aber es erscheint kein Modell.

Abhilfe

- Stellen Sie sicher, dass die Kinematikachsen referenziert sind und dass die Kinematik-Konfiguration erfolgreich ausführbar ist. Eventuelle Fehlermeldungen mit Hinweisen auf Fehlerursachen erscheinen im MessageHandler (im Kontextmenü der MOVI-PLC® im Gerätebaum).

20211732 / DE – 07/2014

14.2 Lastvorsteuerung

14.2.1 Durchsacken der Achsen

Problem

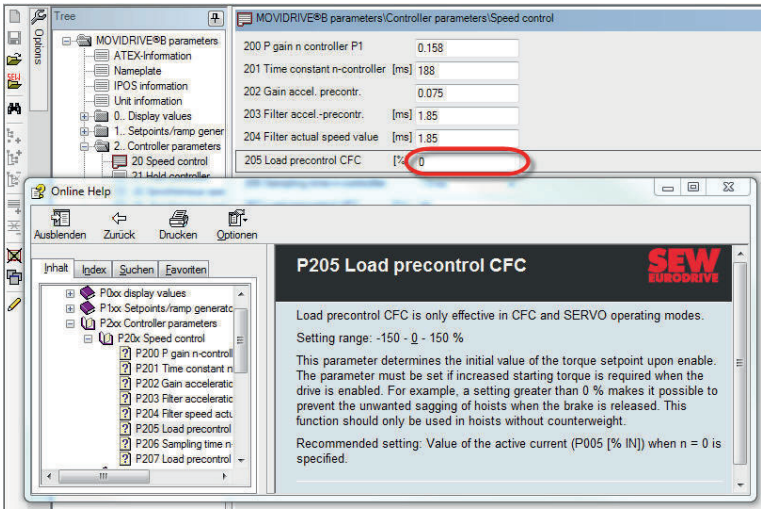
Beim Einschalten der Achsen nach NotAus oder Inverter Inhibit „sacken“ die Achsen durch.

Abhilfe

Um ein „Absacken“ einer unter Last stehenden Achse nach InverterInhibit bzw. SafeTorqueOff zu verhindern, können Sie eine Lastvorsteuerung einstellen.

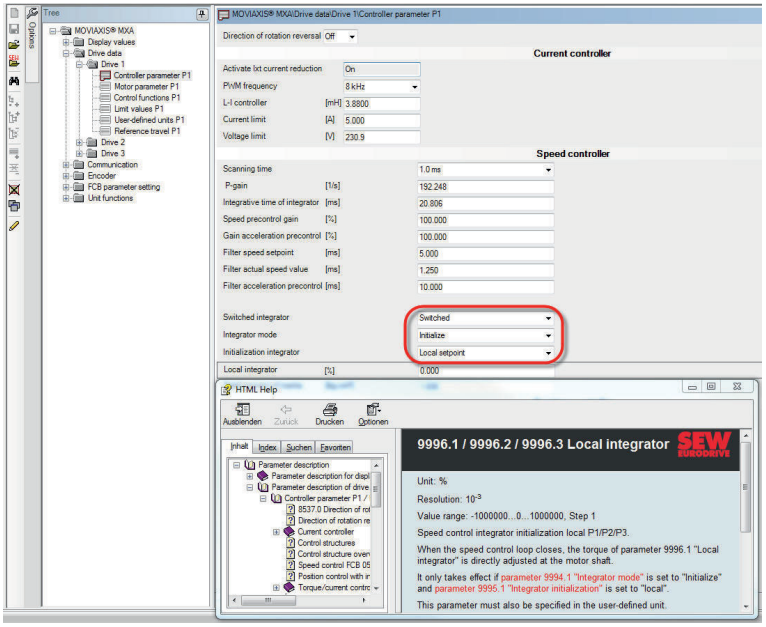
- Parametrieren Sie die Lastvorsteuerung im Parameterbaum von MOVITOOLS® MotionStudio für das betreffende Gerät.

MOVIDRIVE®:


MOVIDRIVE®B parameters\Controller parameters\Speed control
200 P gain n controller P1 0.158
201 Time constant n-controller [ms] 188
202 Gain accel. precontr. 0.075
203 Filter accel.-precontr. [ms] 1.85
204 Filter actual speed value [ms] 1.85
205 Load precontrol CFC [%] 0
Online Help
Ausblenden
Zurück
Drucken
Optionen
Inhalt
Index
Suchen
Favoriten
P205 Load precontrol CFC
Load precontrol CFC is only effective in CFC and SERVO operating modes.
Setting range: -150 - 0 - 150 %
This parameter determines the initial value of the torque setpoint upon enable. The parameter must be set if increased starting torque is required when the drive is enabled. For example, a setting greater than 0 % makes it possible to prevent the unwanted sagging of hoists when the brake is released. This function should only be used in hoists without counterweight.
Recommended setting: Value of the active current (P005 [% IN]) when n = 0 is specified.

9308586123

MOVIAXIS®:



9310111627

14.3 Parametrierung der Umrichter

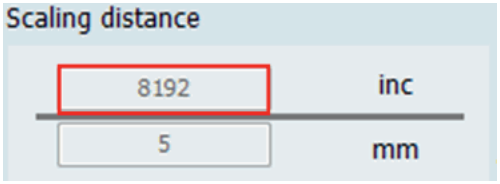
Die Parametrierung der Umrichter erfolgt durch Ausführung der Wizards

- „DriveStartup für MOVI-PLC“ (siehe Schritt 4: Einzelachsen in Betrieb nehmen (→ 🗎 30))
- sowie „Technology Editor MultiMotion Light (oder MultiMotion)” / “Technologiemodul Kinematics” (siehe Schritt 3: Integration des Technologiemoduls (→ 🗎 28))

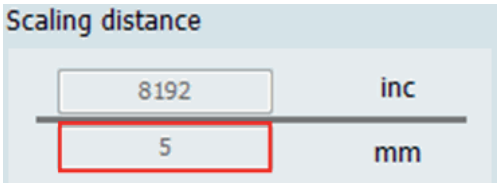
Für Transparenz in speziellen Anwendungen und Situationen illustriert dieses Kapitel für einige ausgewählte Konfigurations-Einstellungen, in welche MultiMotion- und Umrichterparameter die eingestellten Werte übertragen werden.

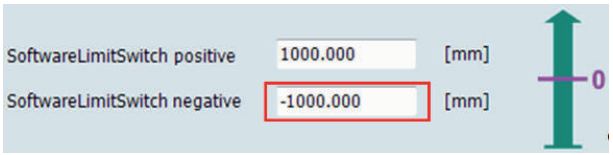
Die Werte der folgenden Variablen der Kinematik-Konfiguration (Datei AppConfigA(n).xml) werden in die entsprechenden Variablen der MultiMotion-Konfiguration (Dateien AxisConfigA(n).xml .. AxisConfigA(n+7).xml) übertragen, sind also in den Konfigurations-Dateien des Techologiemoduls Kinematik sowie der entsprechenden MultiMotion-Achsen enthalten:

- AxisGroupKin.Inst[..].Config.Axis.Numerator[1..8]
- AxisGroupKin.Inst[..].Config.Axis.Denominator[1..8]
- AxisGroupKin.Inst[..].Config.Axis.LimitSwitchNegative[1..8]
- AxisGroupKin.Inst[..].Config.Axis.LimitSwitchPositive[1..8]
- AxisGroupKin.Inst[..].Config.Axis.RapidDeceleration[1..8]

<table>
<tr><td>Kinematik</td><td colspan="2">AxisGroupKin.Inst[..].Config.Axis.Numerator[1..8]
 
</td></tr>
<tr><td>MultiMotion</td><td colspan="2">AxisInterface.Axis[1..8].Config.General.UserUnits.Numerator</td></tr>
<tr><td rowspan="4">MOVIAXIS®</td><td colspan="2">Anwendereinheiten werden im MX immer folgendermaßen belegt:</td></tr>
<tr><td>Position Zähler: 1</td><td>9543.1</td></tr>
<tr><td>Geschwindigkeit Zähler: 1000</td><td>9536.1</td></tr>
<tr><td>Beschleunigung Zähler: 100</td><td>9550.1</td></tr>
<tr><td>MOVIDRIVE®</td><td>Wird bei MDX in keinen Parameter übertragen.</td><td>-</td></tr>
</table>

20211732 / DE – 07/2014

Kinematik	*AxisGroupKin.Inst[..].Config.Axis.Denominator[1..8]* 	
MultiMotion	*AxisInterface.Axis[1..8].Config.General.UserUnits.Denominator*	
MOVIAXIS®	Anwendereinheiten werden im MX immer folgendermaßen belegt:	
	Position Nenner: 1	9544.1
	Geschwindigkeit Nenner: 1	9537.1
	Beschleunigung Nenner: 1	9551.1
MOVIDRIVE®	Wird bei MDX in keinen Parameter übertragen.	-

Kinematik	*AxisGroupKin.Inst[..].Config.Axis.LimitSwitchNegative[1..8]* 	
MultiMotion	*AxisInterface.Axis[1..8].Config.General.LimitSwitch.Negative*	
MOVIAXIS®	Wird bei MX in keinen Parameter übertragen.	-
MOVIDRIVE®	Wird bei MDX in keinen Parameter übertragen.	-

Kinematik	*AxisGroupKin.Inst[..].Config.Axis.LimitSwitchPositive[1..8]* 	
MultiMotion	*AxisInterface.Axis[1..8].Config.General.LimitSwitch.Positive*	
MOVIAXIS®	Wird bei MX in keinen Parameter übertragen.	-
MOVIDRIVE®	Wird bei MDX in keinen Parameter übertragen.	-

20211732 / DE – 07/2014

Kinematik	*AxisGroupKin.Inst[..].Config.Axis.RapidDeceleration[1..8]* 	
MultiMotion	*AxisInterface.Axis[1..8].Config.General.AxisLimits.EnableStopDeceleration* → **A** **Hinweis:** Es ist bei Verwendung des Technologiemoduls Kinematics nicht vorgesehen, die Achsen mit den MultiMotion Stopprampen (also durch Rücksetzen des Eingangssignals *AxisGroupKin.Inst[..].In.General.Enable_Stop*) unabhängig voneinander abzubremsen, da es hierbei i. a. zu einer Bahnabweichung kommt. Statt dessen sollte das Abbremsen durch Rücksetzen des Eingangssignals *AxisGroupKin.Inst[..].In.General.Kin.FeedEnable* erfolgen. Sehr wohl ist es möglich, durch Rücksetzen des Eingangssignals *AxisGroupKin.Inst[..].In.General.Enable_RapidStop* achsweise mit den Notstopp-Rampen abzubremsen.	
	AxisInterface.Axis[1..8].Config.General.AxisLimits.EnableRapidStopDeceleration → **B**	
MOVIAXIS®	**A** → Applikationsgrenze Maximale Beschleunigung	9571.1
	A → Applikationsgrenze Maximale Verzögerung	9572.1
	B → Notstopp (Maximale) Verzögerung	9576.1
MOVIDRIVE®	**A** → System limit max deceleration [ms] - P131 t11 down CW	8471.0
	A → System limit max deceleration [ms] - P133 t11 down CCW	8473.0
	A → System limit max acceleration [ms] - P130 t11 up CW	8470.0
	A → System limit max acceleration [ms] - P132 t11 up CCW	8472.0
	B → Application limit max deceleration [ms] - P136 t13	8476.0
	B → Emergency stop max deceleration [ms] - P137 t14	8477.0

20211732 / DE – 07/2014

Im Folgenden sind einige ausgewählte Parameter aufgelistet, die durch das Technologiemodul Kinematics nicht konfiguriert werden. In die MultiMotion-/Umrichter-Parameter wird der Wert 0 übertragen.

Kinematik	Nicht vorhanden. → In die MultiMotion-/Umrichter-Parameter wird der Wert 0 übertragen.	
MultiMotion	SystemMaxVelocity	
MOVIAXIS®	Systemgrenze Maximale Geschwindigkeit Positiv	9579.1
	Systemgrenze Maximale Geschwindigkeit Negativ	9579.10
	Applikationsgrenze Maximale Geschwindigkeit Positiv	9716.1
	Applikationsgrenze Maximale Geschwindigkeit Negativ	9716.10
MOVIDRIVE®	System limit max velocity [RPM] - P302	8517.0

Kinematik	Nicht vorhanden. → In die MultiMotion-/Umrichter-Parameter wird der Wert 0 übertragen.	
MultiMotion	SystemMaxAcceleration	
MOVIAXIS®	Systemgrenze Maximale Beschleunigung	9573.1
MOVIDRIVE®	Wird bei MDX in keinen Parameter übertragen.	-

Kinematik	Nicht vorhanden. → In die MultiMotion-/Umrichter-Parameter wird der Wert 0 übertragen.	
MultiMotion	SystemMaxDeceleration	
MOVIAXIS®	Systemgrenze Maximale Verzögerung	9574.1
MOVIDRIVE®	Wird bei MDX in keinen Parameter übertragen.	-

20211732 / DE – 07/2014

15 Fehlercodes

15.1 Allgemeine Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_GENERAL_INTERNAL_ERROR	Internere Fehler, bitte SEW-Service kontaktieren.	FD0000
E_KIN_GENERAL_LICENSE_MODE	Bei der Konfiguration wurden keine Lizenzpunkte für die Ausführung der angeforderten Betriebsart konsumiert.	FD0008
E_KIN_GENERAL_MODE	Angeforderte Betriebsart ist in der Situation nicht zulässig.	FD0010
E_KIN_GENERAL_DIRKIN	Fehler bei der Ausführung der direkten kinematischen Trans-formation	FD0020
E_KIN_GENERAL_INVKIN	Fehler bei der Ausführung der inversen kinematischen Transformation	FD0030
E_KIN_GENERAL_AXIS_COUPLING	Fehler bei der Kopplung mehrere Achswerte der Kinematik zu Motor-positionen.	FD0038
E_KIN_GENERAL_WORK-SPACE_NO_AUTOGOON	Verlassen des zulässigen Arbeitsraums, Details in MessageHandler sowie in Out-Variablen	FD0040
E_KIN_GENERAL_MOTORSPEED_LI-MIT_NO_AUTOGOON	Überschreitung der zulässigen Motorgeschwindigkeit, Details in MessageHandler sowie in Out-Variablen	FD0050
E_KIN_GENERAL_AXISINCRE-MENTS_TOOLARGE_NEG	Achsinkremente in der Nähe des negativen 32-Bit-Grenzwerts.	FD0060
E_KIN_GENERAL_AXISINCRE-MENTS_TOOLARGE_POS	Achsinkremente in der Nähe des positiven 32-Bit-Grenzwerts.	FD0070
E_KIN_GENERAL_MESSAGEBUF-FER_CORRUPT	*MessageBuffer* beschädigt durch unerlaubtes Überschreiben	FD0080
E_KIN_GENERAL_MESSAGEBUF-FER_FULL	*MessageBuffer* komplett gefüllt durch die aktuelle -Fehlerserie.	FD0090
E_KIN_GENE-RAL_ABC_STEP_AT_COORD-SYS_CHANGE	Sprung der ABC-Orientierungswerte während des Wechsels -zwischen Koordinatensystemen aufgrund einer ABC-Singularität in der Stellung.	FD00A0
E_KIN_GENE-RAL_ABC_STEP_AT_JOG_CART	Sprung der ABC-Orientierungswerte in der Betriebsart KIN_JOG_CART aufgrund einer ABC-Singularität in der Stellung.	FD00B0
E_KIN_GENERAL_ABC_STEP_AT_TAR-GET_CART	Sprung der ABC-Orientierungswerte in der Betriebsart KIN_TARGET_CART aufgrund einer ABC-Singularität in der Stellung.	FD00C0
E_KIN_GENERAL_TEST_ENABLE	Funktion nicht zulässig ohne Test-Aktivierung	FD0100
E_KIN_GENERAL_AXISCARTBIJECTI-ON_OUT_OF_AXIS_SWLS	Die eineindeutige Zuordnung von Achswerten und kartesischen Koordinaten ist verletzt wegen Verlassens der Axis.SWLS, bitte verfahren Sie die Achsen mittels KIN_JOG/TARGET_AXIS in den zulässigen Arbeitsraums.	FD0110

20211732 / DE – 07/2014

Fehler	Bedeutung	Hex-Code
E_KIN_GENERAL_AXISCARTBIJECTI-ON_TRANSFORM	Die eineindeutige Zuordnung von Achswerten und kartesischen Koordinaten ist verletzt wegen einer unzulässigen Transformation.	FD0120

20211732 / DE – 07/2014

15.2 Konfigurations-Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_CONFIG_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren.	FD1000
E_KIN_CONFIG_NO_CONFIG	Die eingelesene Kinematik-Konfiguration ist nicht korrekt.	FD1008
E_KIN_CONFIG_IN_PROCESS	Während der Übernahme einer Kinematik-Konfiguration ist kein Trigger zum erneuten Konfigurieren zulässig.	FD1010
E_KIN_CONFIG_CYCLE_TIME	Die konfigurierte *CycleTime* ist ≤ 0 ms.	FD1020
E_KIN_CONFIG_LICENSE_GET	Die den konfigurierten Funktionen entsprechende Anzahl an Lizenzpunkten kann nicht konsumiert werden, da die Speicherkarte nicht genug Lizenzpunkte hat, oder weil eine andere Technologiefunktion bereits Lizenzpunkte konsumiert hat, so dass zu wenig Lizenzpunkte verblieben sind.	FD1030
E_KIN_CONFIG_LICENSE_CORRUPT	Die Lizenzprüfung ist beschädigt.	FD1040
E_KIN_CONFIG_LICENSE_KINTYPE	In der Konfiguration ist die Lizenzierung der gewählten Kinematik nicht angefordert.	FD1050
E_KIN_CONFIG_LICENSE_COORD-SYS	In der Konfiguration ist die Lizenzierung der Nutzung von WCS, PCS1, PCS2 nicht angefordert.	FD1060
E_KIN_CONFIG_KINTYPE	Die konfigurierte Kinematik wird nicht unterstützt.	FD1070
E_KIN_CONFIG_DIRKIN	Fehler bei der Ausführung der direkten kinematischen Transformation während der Übernahme der Konfiguration.	FD1080
E_KIN_CONFIG_AXIS_DECOUPLING	Fehler bei der Entkopplung mehrerer Motorpositionen zu Achswerten der Kinematik	FD1088
E_KIN_CONFIG_KINPAR_USE	Fehlerhafte Konfiguration der Verwendung von Parametern, z. B. KIN_USE_DEGR für eine Linearachse	FD1090
E_KIN_CONFIG_KIN-PAR_OUT_OF_RANGE	Fehlerhafte Parameterwerte, z. B. negative Armlänge	FD10A0
E_KIN_CONFIG_KINLIMIT_USE	Fehlerhafte Einstellung der Verwendung einer Kinematik-Limitierung, z. B. KIN_USE_NONE für eine im gewählten Kinematiktyp vorhandene Limitierung	FD10B0
E_KIN_CONFIG_KINLIMIT_MINMAX	Fehlerhafte Einstellung der Software-Endschalter der Kinematik-Limitierungen, z. B. SWLS_Neg > SWLS_Pos	FD10C0
E_KIN_CONFIG_AXIS_USE	Fehlerhafte Konfiguration der Verwendung einer Achse, z. B. *-AxisUse* = KIN_USE_NONE für eine im gewählten Kinematiktyp vorhandene Achse	FD10F0
E_KIN_CONFIG_AXISPOS_MINMAX	Fehlerhafte Einstellung der Software-Endschalter für eine Achse, z. B. SWLS_Neg > SWLS_Pos	FD1100
E_KIN_CONFIG_AXIS_JOGVEL100	Fehlerhafte Einstellung von *AxisJogVel100Percent ≤ 0*	FD1130
E_KIN_CONFIG_AXIS_JOGACCDEC	Fehlerhafte Einstellung von *AxisJogAccDec ≤ 0*	FD1138
E_KIN_CONFIG_AXIS_RAPIDDEC	Fehlerhafte Einstellung von *AxisRapidDeceleration ≤ 0*	FD1140

20211732 / DE – 07/2014

Fehler	Bedeutung	Hex-Code
E_KIN_CONFIG_AXIS_RAPIDJERK	Fehlerhafte Einstellung von *AxisRapidJerkTime ≤ 0*	FD1150
E_KIN_CONFIG_NUMERATOR	Fehlerhafte Einstellung von Numerator = 0	FD1160
E_KIN_CONFIG_DENOMINATOR	Fehlerhafte Einstellung von Denominator = 0	FD1170
E_KIN_CONFIG_MOTORSPEED_CONVERSION	Fehlerhafte Einstellung von *MotorSpeedConversionFactor ≤ 0*	FD1180
E_KIN_CONFIG_MOTORSPEED_MAX	Fehlerhafte Einstellung von *MotorSpeedMaxLimit ≤ 0*	FD1190
E_KIN_CONFIG_MOTORSPEED_WARN_PERC	Fehlerhafte Einstellung von *MotorSpeedWarningPercentage < 1*	FD11A0
E_KIN_CONFIG_MOTORSPEED_LIMIT_PERC	Fehlerhafte Einstellung von *MotorSpeedLimitPercentage < MotorSpeedWarningPercentage*	FD11B0
E_KIN_CONFIG_CARTUNIT_USE	Fehlerhafte Einstellung von -*CartUnitUse*, z. B. KIN_USE_DEGR für eine Translations-Dimension	FD11C0
E_KIN_CONFIG_CART_JOGLEVEL100	Fehlerhafte Einstellung von *CartJogVel100Percent ≤ 0*	FD11D0
E_KIN_CONFIG_CART_JOGACCDEC	Fehlerhafte Einstellung von *CartJogAccDec ≤ 0*	FD11D8
E_KIN_CONFIG_CART_RAPIDDEC	Fehlerhafte Einstellung von *CartRapidDeceleration ≤ 0*	FD11E0
E_KIN_CONFIG_CART_RAPIDJERK	Reserviert	FD11F0
E_KIN_CONFIG_CARTPOS_MINMAX	Fehlerhafte Einstellung der Software-Endschalter für den -kartesischen Arbeitsraum in KCS, z. B. SWLS_Neg > SWLS_Pos	FD1200
E_KIN_CONFIG_CART_MODULO_USE	Fehlerhafte Einstellung von *CartModuloUse*, z. B. TRUE für eine Translations-Dimension	FD1230
E_KIN_CONFIG_CART_MODULO_RANGE	Fehlerhafte Einstellung von *CartModuloUnderfollow/Overflow*; der Bereich stellt kein ganz-zahliges Vielfaches von 2*PI bzw. 360° dar	FD1240
E_KIN_CONFIG_CP_RAPIDTRANSDEC	Fehlerhafte Einstellung von *CpRapidTransDeceleration ≤ 0*	FD1250
E_KIN_CONFIG_CP_RAPIDTRANSJERK	Reserviert	FD1260
E_KIN_CONFIG_AXISCARTBIJECTION_NOT_WITH_MAPABC	*AxisGroupKin.Inst[..].Config.General.AxisCartBijection* and MapABCToStartupValuesAtConfig MUST NOT be TRUE both	FD1270
E_KIN_CONFIG_AXISCARTBIJECTION_MODEL_NOT_SUPPORTED	*AxisGroupKin.Inst[..].Config.General.AxisCartBijection* not supported for the configured kinematics model	FD1280
E_KIN_CONFIG_AXISCARTBIJECTION_AXIS_SWLS_TOO_LARGE	*AxisGroupKin.Inst[..].Config.General.AxisCartBijection* not possible because of too large Axis.SWLS areas	FD1290
E_KIN_CONFIG_AXISCARTBIJECTION_CART_OFFSET_DIRREV	*AxisGroupKin.Inst[..].Config.General.AxisCartBijection* not possible because of Cart.Offset_KCS or DirectionReversal_KCS <> 0	FD12A0
E_KIN_CONFIG_CART_MODULO_NOT_WITH_AXISCARTBIJECTION	Cart.ModuloUse has to be FALSE, da General.AxisCartBijection is TRUE	FD1238

20211732 / DE – 07/2014

15.3 Allgemeine Parameter-Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_GENPAR_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren.	FD2000
E_KIN_GENPAR_OVERRIDE_CHANGE	Änderung von *Override*, während *Standstill.ActCoordSys* = *FALSE*; der neue Override wird bei Erreichen von Stillstand übernommen.	FD2010
E_KIN_GENPAR_OVERRIDE_SMALL	*Override* = *0* nicht zulässig	FD2018
E_KIN_GENPAR_OVERRIDE_LARGE	*AxisGroupKin.Inst[..].In.General.Kin.OverridePercentage* > *AxisGroupKin.Inst[..].Config.General.MaxOverridePercentage* nicht zulässig.	FD2020
E_KIN_GENPAR_JOGPERC_LARGE	*AxisGroupKin.Inst[..].In.Jog.VelocityPercentage* > *AxisGroupKin.Inst[..].Config.General.MaxJogVelocityPercentage* nicht zulässig.	FD2030
E_KIN_GENPAR_COORDSYS	Der Wechsel des Koordinatensystems ist in der Situation nicht zulässig und wird auch nicht ausgeführt.	FD2040
E_KIN_GENPAR_LICENSE_COORDSYS	In der Konfiguration ist die Lizenzierung der Nutzung von WCS, PCS1, PCS2 nicht angefordert.	FD2050
E_KIN_GENPAR_MODULO_MODE	*AxisGroupKin.Inst[..].In.Target.ModuloMode* = *KIN_DIRECTION_NIL* nicht zulässig für eine kartesische Dimension im Modulo-Betrieb.	FD2060

20211732 / DE – 07/2014

15.4 Target Parameter-Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_TARGPAR_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren.	FD3000
E_KIN_TARGPAR_VEL	Fehlerhafte Einstellung von *AxisGroupKin.Inst[..].In.Target.Axis / Cart.Velocity < 0*.	FD3008
E_KIN_TARGPAR_ACC	Fehlerhafte Einstellung von *AxisGroupKin.Inst[..].In.Target.Axis / Cart.Acceleration ≤ 0*.	FD3010
E_KIN_TARGPAR_DEC_SMALL	Fehlerhafte Einstellung von *AxisGroupKin.Inst[..].In.Target.Axis / Cart.Deceleration ≤ 0*.	FD3020
E_KIN_TARGPAR_DEC_LARGE	Fehlerhafte Einstellung von *AxisGroupKin.Inst[..].In.Target.Axis / Cart.Deceleration > AxisGroupKin.Inst[..].Config.Axis / Cart.RapidDeceleration*, abhängig von der aktuellen Betriebsart KIN_TARGET_AXIS / CART.	FD3030
E_KIN_TARGPAR_JERK	Fehlerhafte Einstellung von *AxisGroupKin.Inst[..].In.Target.Axis / Cart.Jerk < AxisGroupKin.Inst[..].Config.Axis / Cart.RapidJerk*, abhängig von der aktuellen Betriebsart KIN_TARGET_AXIS / CART.	FD3040
E_KIN_TARGPAR_MODULO_OUT_OF_RANGE	Fehlerhafte Einstellung von *AxisGroupKin.Inst[..].In.Target.Position > Cart.ModuloUnderflow* oder *AxisGroupKin.Inst[..].In.Target.Axis / Cart.Velocity > CartModulo.Overflow*.	FD3050
E_KIN_TARGPAR_OUT_OF_AXIS_SWLS	TARGET_AXIS zu kartesischer Zielposition nicht möglich, da die der Zielposition entsprechenden Achswerte ausserhalb der Achsen-SWLS liegen.	FD3060
E_KIN_TARGPAR_AXISCARTBIJECTION_NOT_WITH_MAPABC	*AxisGroupKin.Inst[..].Config.General.AxisCartBijection* und *AxisGroupKin.Inst[..].In.Target.ABCMapping* dürfen nicht beide auf TRUE gesetzt sein.	FD3070

20211732 / DE – 07/2014

15.5 ContinuousPath Parameter-Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_CPPAR_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren	FD4000
E_KIN_CPPAR_LICENSE_PLUS-TARGCART	In der Konfiguration ist die Lizenzierung der Nutzung von KIN_TARGET_CART für die Umorientierung des TCP während der Ausführung einer Translation mittels ContinuousPath nicht ange-fordert.	FD4010
E_KIN_CPPAR_UNCLEAR_PLUS-TARGCART_OR_INCLUDINGABC	Keine klare Auswahl, ob ABC während ContinuousPath synchronisiert interpoliert werden soll – oder als Parallelprozess mittels TARGET_CART	FD4018
E_KIN_CPPAR_BACKTO-PATH_AXIS_COUPLING	Fehler bei der Kopplung mehrerer Achswerte der Kinematik zu Motorpositionen während einer CP BackTo-Path-Bewegung.	FD4028
E_KIN_CPPAR_BACKTO-PATH_AXIS_DECOUPLING	Fehler bei der Entkopplung mehrerer Motorpositionen zu Achswerten der Kinematik während einer CP Back-ToPath-Bewegung.	FD4029
E_KIN_CPPAR_BACKTOPATH_MODE	Kein Zurückpositionieren auf die CP-Bahn, da keine ContinuousPath Betriebsart aktiv ist.	FD4030
E_KIN_CPPAR_BACKTOPATH_DIST	Kein Zurückpositionieren auf die ContinuousPath Bahn, da *AxisGroupKin.Inst[..].In.Cp.BackTo-Path.MaxInitialDistance* < Distanz zu Beginn der Rückpositionierung.	FD4040
E_KIN_CPPAR_PATH_FIDELITY	Fehlerhafte Einstellung von *AxisGroup-Kin.Inst[..].In.Cp.Path.FidelityPercentage* < *90* oder > 100.	FD4050
E_KIN_CPPAR_TRANSPERC_LARGE	*In.Cp.Path.VelocityPercentage* > *Config.Cp. MaxPath-VelocityPercentage*	FD4060
E_KIN_CPPAR_MASTERPOS	Der Werteverlauf am Eingang *AxisGroup-Kin.Inst[..].In.MasterPosition.Cp* darf nicht abfallen.	FD4070
E_KIN_CPPAR_DIRKIN	ContinuousPath Segment nicht übernommen wegen eines Fehlers bei der Ausführung der direkten kinematischen Transformation.	FD4080
E_KIN_CPPAR_TRANSVEL	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Velocity* ≤ *0*.	FD4090
E_KIN_CPPAR_TRANSACC	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Acceleration* ≤ *0*.	FD40A0
E_KIN_CPPAR_TRANSDEC_SMALL	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Deceleration* ≤ *0*.	FD40B0
E_KIN_CPPAR_TRANSDEC_LARGE	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Deceleration* > *Config.CpRapidTransDeceleration*.	FD40C0
E_KIN_CPPAR_TRANSJERK	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Jerk* < *Config.Cp.RapidTransJerk*.	FD40D0

20211732 / DE – 07/2014

Fehler	Bedeutung	Hex-Code
E_KIN_CPPAR_QUEUE_SIZE	ContinuousPath Segment nicht übernommen, da die interne Queue komplett gefüllt und die zugehörige Bahn noch nicht abgefahren ist.	FD40E0
E_KIN_CPPAR_QUEUE_SIZE_REVERSE	Moving reversely close to the queue limitation -> abrupt stop	FD40E8
E_KIN_CPPAR_QUEUE_NOT_FILL_AT_ERROR	Continuous Path Segment nicht übernommen wegen aktuell vorliegendem Kinematik-Fehler.	FD40F0
E_KIN_CPPAR_PATH_LENGTH	ContinuousPath Segment nicht übernommen, da die insgesamt abgefahrene und geplante Bahnlänge seit dem letzten Leeren der Queue den maximal zulässigen Wert überschreitet.	FD4100
E_KIN_CPPAR_LIN_DIST	LIN-Segment nicht übernommen wegen der Länge 0.	FD4110
E_KIN_CPPAR_LIN_BLENDING_DIST	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Translation.Deceleration ≤ 0*.	FD4120
E_KIN_CPPAR_LIN_BLENDING_MAXPERC	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Blending.LimitationPercentage < 1 oder < 99*.	FD4130
E_KIN_CPPAR_CIRC_MODE	CIRC-Segment nicht übernommen wegen eines in der Situation nicht zulässigen *AxisGroupKin.Inst[..].In.Cp.Segment.CircMode*.	FD4140
E_KIN_CPPAR_CIRC_PLANE	CIRC-Segment nicht übernommen, da die Endtangente des vorhergehenden ContinuousPath Segments nicht in der Kreisebene liegt.	FD4150
E_KIN_CPPAR_CIRC_ANGLE	CIRC-Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.CircAngle ≤ 0*.	FD4160
E_KIN_CPPAR_CIRC_RADIUS	CIRC-Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Radius ≤ 0* bei der eingestellten CIRC-Betriebsart nicht zulässig.	FD4170
E_KIN_CPPAR_CIRC_CENTER_CORRECTION	Reserviert.	FD4180
E_KIN_CPPAR_CIRC_DIRECTION	CIRC-Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.CircDirection* = KIN_DIRECTION_NIL / SHORT	FD4190
E_KIN_CPPAR_CIRC_DISTORTION	CIRC-Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Circ.EllipticDistortion ≤ 0*.	FD41A0
E_KIN_CPPAR_ROTATION_VEL	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Rotation.Velocity[j]* <= 0, j = 4..6	FD4320
E_KIN_CPPAR_ROTATION_ACC	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Acceleration[j]* <= 0, j = 4..6	FD4330
E_KIN_CPPAR_ROTATION_DEC_SMALL	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Deceleration[j]* <= 0, j = 4..6	FD4340

20211732 / DE – 07/2014

Fehler	Bedeutung	Hex-Code
E_KIN_CPPAR_ROTATI-ON_DEC_LARGE	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Carttion.Deceleration[j] > AxisGroupKin.Inst[..].Config.Cart.RapidDeceleration[j], j = 4..6*	FD4350
E_KIN_CPPAR_ROTATION_JERK	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Cart.Jerk < AxisGroupKin.Inst[..].Config.Cart.RapidJerk[j], j = 4..6*	FD4360
E_KIN_CPPAR_AXIS_VEL	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Velocity[j]* <= 0	FD43B0
E_KIN_CPPAR_AXIS_ACC	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Acceleration[j]* <= 0	FD43C0
E_KIN_CPPAR_AXIS_DEC_SMALL	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Deceleration[j]* <= 0	FD43D0
E_KIN_CPPAR_AXIS_DEC_LARGE	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Deceleration[j] > AxisGroupKin.Inst[..].Config.Axis.RapidDeceleration[j]*	FD43E0
E_KIN_CPPAR_AXIS_JERK	ContinuousPath Segment nicht übernommen, da *AxisGroupKin.Inst[..].In.Cp.Segment.Axis.Jerk < AxisGroupKin.Inst[..].Config.Axis.RapidJerk[j]*	FD43F0

20211732 / DE – 07/2014

15.6 Profil-Generator Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_PROFGEN_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren.	FD5000
E_KIN_PROFGEN_INIT_DELTAT	Fehlerhafte Einstellung von *KinProfGen.Init.DeltaT < 0*.	FD5010
E_KIN_PROFGEN_INIT_MODULO_RANGE	Fehlerhafte Einstellung von *KinProfGen.Init.ModuloUnderflow ≤ Overflow.*	FD5020
E_KIN_PROFGEN_INIT_OUT_OF_MODULO_RANGE	Fehlerhafte Einstellung von *KinProfGen.Init.InitPos* außerhalb des Bereichs *KinProfGenInit.ModuloUnderflow (Overflow)*.	FD5030
E_KIN_PROFGEN_MODULO_TARGET_OUT_OF_RANGE	Fehlerhafte Einstellung von *KinProfGen.TargPos* außerhalb des Bereichs *KinProfgenInit.ModuloUnderflow (Overflow)*.	FD5040
E_KIN_PROFGEN_MODULO_MODE	Fehlerhafte Belegung von *KinProfGen.ModuloMode = KIN_DIRECTION_NIL*	FD5050
E_KIN_PROFGEN_MODULO_TOO_FAST_POSITIVE	Mehr als der komplette Modulo-Bereich müsste in einem Zyklus in positiver Richtung durchlaufen werden.	FD5060
E_KIN_PROFGEN_MODULO_TOO_FAST_NEGATIVE	Mehr als der komplette Modulo-Bereich müsste in einem Zyklus in negativer Richtung durchlaufen werden.	FD5070
E_KIN_PROFGEN_VEL_MAX	Fehlerhafte Belegung von *KinProfGen.Vmax < 0*.	FD5080
E_KIN_PROFGEN_ACC_MAX	Fehlerhafte Belegung von *KinProfGen.Amax ≤ 0*.	FD5090
E_KIN_PROFGEN_DEC_MAX	Fehlerhafte Belegung von *KinProfGen.Dmax ≤ 0*.	FD50A0
E_KIN_PROFGEN_DEC_RAPID	Fehlerhafte Belegung von *KinProfGen.D_Rapid ≤ 0*.	FD50B0
E_KIN_PROFGEN_JERK	Fehlerhafte Belegung von *KinProfGen.JerkTime < 0*.	FD50C0
E_KIN_PROFGEN_JERK_RAPID	Fehlerhafte Belegung von *KinProfGen.JerkTime_Rapid < 0*.	FD50D0

20211732 / DE – 07/2014

15.7 3D-Simulation Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_SIMU3D_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren.	FDA000
E_KIN_SIMU3D_LICENSE_GET	Die für 3D-Simulation erforderlichen 10 Lizenzpunkte können nicht konsumiert werden, da die Speicherkarte nicht genug Lizenzpunkte hat, oder weil eine andere Technologiefunktion bereits Lizenzpunkte konsumiert hat, so dass zu wenig Lizenzpunkte verblieben sind.	FDA010
E_KIN_SIMU3D_LICENSE_CORRUPT	Die Lizenzprüfung ist beschädigt.	FDA020
E_KIN_SIMU3D_CONNECT_TCP		FDA030
E_KIN_SIMU3D_CONNECT_UDP		FDA040
E_KIN_SIMU3D_BUILDING_SCENE		FDA050
E_KIN_SIMU3D_REG_CYCLIC_TELEGR		FDA060
E_KIN_SIMU3D_RECV_TELEGR_ID_TIMEOUT		FDA061
E_KIN_SIMU3D_RECV_TELEGR_ID		FDA062
E_KIN_SIMU3D_SEND_CYCLIC_TELEGR_TCP		FDA070
E_KIN_SIMU3D_SEND_CYCLIC_TELEGR_UDP		FDA071

20211732 / DE – 07/2014

15.8 AxisGroupControl Kinematik Fehler

Fehler	Bedeutung	Hex-Code
E_KIN_AXISGROUPKIN_INTERNAL_ERROR	Interner Fehler, bitte SEW-Service kontaktieren.	FDF000
E_KIN_AXISGROUPKIN_READING_AXIS_VALUES_NOT_POSSIBLE	Lesen der Inkremente von den -Achsen war nicht erfolgreich. Prüfen Sie, ob der Umrichter bereit ist.	FDF010
E_KIN_AXISGROUPKIN_WRITING_AXIS_VALUES_NOT_POSSIBLE	Schreiben der Startwerte für den Baustein MC_KinControl war nicht erfolgreich.	FDF020
E_KIN_AXISGROUPKIN_CONFIGURATION_PRG_DONE_NOT_FALSE	Handshake zum Konfigurations-programm war nicht erfolgreich.	FDF030
E_KIN_AXISGROUPKIN_CONFIGURATION_PRG_DONE_NOT_TRUE		FDF040
E_KIN_AXISGROUPKIN_READ_CONFIG_FROM_SD_CARD_NOT_DONE	Lesen der Konfiguration von der Speicherkarte nicht ausgeführt.	FDF041
E_KIN_AXISGROUPKIN_CONFIG_IS_NOT_TAKEN_OVER	Kinematik-Konfiguration wurde nicht übernommen, z. B. da Kon-figuration fehlerhaft ist oder weil die zyklische Task steht.	FDF050
E_KIN_AXISGROUPKIN_CLEAR_LAG_FALSE_NOT_POSSIBLE	Handshake zum unterlagerten AxisControl fehlerhaft.	FDF060
E_KIN_AXISGROUPKIN_LAG_IS_NOT_CLEARED		FDF070
E_KIN_AXISGROUPKIN_NOT_ALL_AXES_IN_INTERPOLATION_MODE	Eine oder mehrere Achsen konnten nicht in die Interpolations-Betriebsart geschaltet werden (siehe MultiMotion AxisInterface).	FDF080
E_KIN_AXISGROUPKIN_BACKTOPATH_NOT_ACTIVE	Handshake zum MC_KinControl war fehlerhaft.	FDF090
E_KIN_AXISGROUPKIN_BACKTOPATH_CLEAR_LAG_FALSE_NOT_POSSIBLE	Handshake zum unterlagerten Achstreiber fehlerhaft.	FDF0A0
E_KIN_AXISGROUPKIN_BACKTOPATH_LAG_NOT_CLEARED		FDF0B0
E_KIN_AXISGROUPKIN_BACKTOPATH_NOT_IN_INTERPOLATION	Eine oder mehrere Achsen konnten nicht in die Interpolations-Betriebsart geschaltet werden (siehe MultiMotion AxisInterface).	FDF0C0
E_KIN_AXISGROUPKIN_NUMBER_OF_KIN_AXES_NOT_VALID	Fehlerhafte Einstellung der Anzahl von Kinematikachsen ≤ 0 oder > 6.	FDF0D0

20211732 / DE – 07/2014

Fehler	Bedeutung	Hex-Code
E_KIN_AXISGROUPKIN_NUMBER_OF_AUX_AXES_NOT_VALID	Fehlerhafte Einstellung der Anzahl von Hilfsachsen < 0 oder > 2.	FDF0E0
E_KIN_AXISGROUPKIN_MODE_NOT_VALID	Ausgewählte Betriebsart wird vom AxisGroupControl Kinematics nicht unterstützt.	FDF0F0
E_KIN_AXISGROUPKIN_INSTANCEID_NOT_VALID	Die in der AxisGroup-Konfiguration eingestellte InstanceID ist nicht gültig. Die InstanceID darf nicht kleiner als 1 sein und nicht größer als die maximale Anzahl der Instanzen (3 .. 12).	FDF100
E_KIN_AXISGROUPKIN_INSTANCEID_DUPLICATE	Die in der AxisGroup-Konfiguration eingestellte InstanceID wurde bei mindestens einer weiteren Instanz vergeben.	FDF110
E_KIN_AXISGROUPKIN_INSTANCENAME_DUPLICATE	Der in der AxisGroup-Konfiguration eingestellte InstanceName wurde bei mindestens einer weiteren Instanz vergeben.	FDF120
E_KIN_AXISGROUPKIN_NUMBER_OF_INSTANCES_NOT_VALID	*NumberOfInstances* größer als *MaxNumberOfAGKInstances.*	FDF130

20211732 / DE – 07/2014

16 Glossar und Abkürzungsverzeichnis

Im Folgenden finden Sie einen Glossar und ein Abkürzungsverzeichnis, - beides in tabellarischer Form.

Der Glossar enthält die wichtigsten Fachbegriffe aus dem vorliegenden Handbuch und ihre Bedeutung.

Im Abkürzungsverzeichnis finden Sie zu häufig verwendeten englischsprachigen Abkürzungen, den vollständigen Begriff, sowie dessen deutschsprachige Entsprechung (falls vorhanden). Die Bedeutung dazu entnehmen Sie wiederum dem Glossar.

16.1 Glossar

A	Bedeutung
ABC-Darstellung	Drei Winkel zur Beschreibung der Orientierung eines Koordinatensystems in Bezug auf ein Referenzkoordinatensystem. Zur Transformation wird das Referenzkoordinatensystem mit dem Winkel 'A' mathematisch positiv um die Z-Achse gedreht. Anschließend wird mit dem Winkel 'B' um die Y-Achse des verdrehten Koordinatensystems gedreht. Abschließend wird eine Drehung mit dem Winkel 'C' um die X-Achse des erneut verdrehten Koordinatensystems durchgeführt.
Achsengeschwindigkeit	Geschwindigkeit der bis zu 6 kinematischen Achsen und der 2 kinematischen Hilfsachsen (rotatorische Achsen / Linearachsen).
Achsenlimitierungen	Grenzwerte für Achskoordinaten. Bei Überschreitung geht die Kinematiksteuerung i. Allg. in einen Fehlerzustand über.
Achsennullpunkt	Position einer Achse bei 0 Motorinkrementen (bei eineindeutiger Zuordnung eines Motors zu einer Achse).
Achsenpfeil	Pfeil in der 3D-Simulation, welcher den Arbeitsbereich (innerhalb der Software-Endschalter) sowie den Achsennullpunkt (also bei 0 Motorinkremente) und die Richtung anzeigt, in welcher die Motorinkremente zunehmen (bei eineindeutiger Zuordnung eines Motors zu einer Achse).
Achskoordinatensystem (ACS)	Koordinatensystem, in dem die Stellung der Kinematik und der Hilfsachsen durch die Achskoordinaten beschrieben wird.
Achsraum	Mehrdimensionaler mathematischer Raum, der durch die Achsen aufgespannt wird. Interpolation im Achsraum bedeutet, dass für jede Achse ein Profil zu den Zielwerten bestimmt wird.
Arbeitsbereichsüberwachung	Überwachung der Einhaltung konfigurierter Arbeitsraumgrenzen.
Arbeitsraum	Konfigurierter Bereich, innerhalb dessen sich die Kinematik bewegen darf. Er wird eingeschränkt durch das Kinematikmodell, die Kinematiklimitierungen, Achsenlimitierungen und kartesischen Limitierungen.
Armsegment	Ein Abschnitt der kinematischen Kette.
Articulated Kinematik	Konfigurierter Bereich, innerhalb dessen sich die Kinematik bewegen darf. Er wird eingeschränkt durch das Kinematikmodell, die Kinematiklimitierungen, Achsenlimitierungen und kartesischen Limitierungen.

B	Bedeutung
Bahnform	Geometrische Gestalt einer Bahn im kartesischen Raum (z. B. eine Gerade oder ein Kreisbogen).

20211732 / DE – 07/2014

B	Bedeutung
Bahnfortschritt	Durch den Werkzeugmittelpunkt (TCP) der Kinematik zurückgelegte Strecke entlang der vorgegebenen Bahn bzw. durchgeführte Drehung des Werkzeugkoordinatensystems (TCS).
Bahngenauigkeit	Maß für die geometrische Übereinstimmung der Bahn des Werkzeugmittelpunkts (TCP) der Kinematik mit der vorgegebenen Bahn.
Bahngeschwindigkeit	Geschwindigkeit, mit welcher sich der Werkzeugmittelpunkt (TCP) der Kinematik entlang der Bahn bewegt.
Bahninterpolation	Bestimmung von Zwischenpunkten zwischen Start- und Endpunkt auf der vorgegebenen Bahn.
Bahnverlauf	Bewegung des Werkzeugmittelpunkts (TCP) einer Kinematik entlang der vorgegebenen Bahn. Dies beinhaltet sowohl die Bahnform, als auch den Bahnfortschritt als Funktion der Zeit.
Basiskoordinatensystem (BCS)	Bezugskoordinatensystem für die direkte kinematische Transformation, i. d. R. verbunden mit dem Robotersockel .Das Basiskoordinatensystem (BCS) kann in der Kinematikkonfiguration mithilfe der Variablen *Cart.Offset_KCS* gegenüber dem Kinematikkoordinatensystem (KCS) verschoben oder verdreht werden. BCS ist im Gegensatz zu ACS/KCS/WCS/ PCS1/2 nicht zur Bewegungssteuerung auswählbar, wird jedoch in der 3D-Simulation angezeigt.
Befehlsgeber	Bedienelement, mit dem der Mensch Einfluss auf die Anlage / Maschine nehmen kann (z. B. Taster für Tippbetrieb in positive und negtive Richtung).
Bewegungsablaufprogramm	Programm, welches den Ablauf von Bewegungen der Kinematik und weiterer Achsen in die Kinematiksteuerung einspeist.
Bewegungsführung	Steuerung des Bahnverlaufs
Bewegungszustand	Zustand einer Bewegung, der die aktuelle Pose, die Geschwindigkeit sowie die Beschleunigung des Werkzeugmittelpunkts (TCP) und aller Achsen umfasst.
Bibliotheksverwalter	Tool des PLC-Editors, in dem Bibliotheken in das MOVI-PLC®-Projekt eingebunden oder gelöscht werden können.
Blending	siehe "Überschleifen"
Boot-Projekt	MOVI-PLC®-Projekt, welches beim Starten der Motion-Controll-Steuerung automatisch gestartet und ausgeführt wird.

C	Bedeutung
Continuous-Path-Betriebsart	Bahninterpolations-Betriebsart, bei der die Bahn i. Allg. geometrisch im Raum beschrieben wird (z. B. Geraden-/Kreissegmente, sowie geometrisch definierte Überschleifbereiche).
CP-Queue	Abarbeitungsliste von CP-Segmenten in der Kinematiksteuerung

D	Bedeutung
Deltakinematik	Parallelkinematik, die durch eine Dreiecksanordnung kinematischer Teilketten charakterisiert ist.

20211732 / DE – 07/2014

E	Bedeutung
Entfernungsprüfung	Bestimmung der Entfernung zu einem Zielpunkt, entweder auf kürzestem Weg im kartesischen Raum oder entlang eines Bahnverlaufs, sowie Prüfung hinsichtlich Überschreitung eines Grenzwerts. Es kann sich um eine translatorische oder eine rotatorische Entfernung handeln.

F	Bedeutung
Flanschkoordinatensystem (FCS)	Kartesisches Koordinatensystem, welches sich aus dem Basiskoordinatensystem BCS mithilfe der direkten kinematischen Transformation ergibt.
Flanschmittelpunkt (FCP)	Ursprung des Flanschkoordinatensystems der Kinematik.
Freiheitsgrad (DOF)	Frei wählbare, voneinander unabhängige Bewegungsmöglichkeiten der Kinematik.

G	Bedeutung
Geradensegment	Bahnabschnitt in der Betriebsart Continuous Path (CP), welcher einer Geraden im kartesischen Raum entspricht.
Geschwindigkeitsprofil	Verlauf der Geschwindigkeit z. B. einer Achse oder einer kartesischen Koordinate oder des Werkzeugmittelpunkts (TCP) entlang des Bahnverlaufs als Funktion der Zeit.
Glättung	Reduktion einer Unstetigkeit, z. B. eines Krümmungssprungs im Bahnverlauf
Gravitationskoordinatensystem (GCS)	Koordinatensystem, bei dem die Z-Achse im Raum vertikal nach oben zeigt (entgegen der Gravitationskraft der Erde). In der Default-Ansicht der 3D-Simulation zeigt die Z-Achse nach oben und die X-Achse nach rechts. Die Orientierung des Weltkoordinatensystems (WCS) kann in Bezug auf das Gravitationskoordinatensystem (GCS) konfiguriert werden.
Grundstellungsfahrt	Bewegung der Kinematik in eine Grundstellung. Diese kann abhängig von der Applikation achsweise oder kartesisch erfolgen und aus nur einem Bewegungsauftrag oder einer Abfolge von Bewegungen bestehen.

H	Bedeutung
Hexapod-Kinematik	Parallelkinematik, die durch 6 parallele kinematische Ketten charakterisiert ist.

I	Bedeutung
Interpolation	Bestimmung einer bestimmten Bahn zwischen zwei Punkten.
Interpolationstakt	Takt, in dem interpoliert wird, d. h. in dem Interpolationspunkte berechnet werden.

K	Bedeutung
Kartesisch	Kartesisch bezieht sich auf die Koordinaten oder die Bewegung entlang den Richtungsachsen eines kartesischen Koordinatensystems. Bei einem kartesischen Koordinatensystem stehen die Richtungsachsen (z. B. X, Y, Z) senkrecht aufeinander, schneiden sich also im 90°-Winkel.
Kartesische Limitierung	Grenzwerte für die kartesischen Koordinaten des Werkzeugmittelpunkts (TCP). Bei Überschreitung geht die Kinematiksteuerung i. Allg. in einen Fehlerzustand über.
Kartesisches Gantry	Kinematikmodell, bei dem 2 bzw. 3 Linearachsen senkrecht zueinander stehen und damit einen kartesischen Arbeitsraum aufspannen.

20211732 / DE – 07/2014

K	Bedeutung
Kartesische Koordinatensysteme	Bei SEW-EURODRIVE eines der folgenden Koordinatensysteme: KCS, WCS, PCS1, PCS2
Kinematik	Die Kinematik beschreibt die Position und die Bewegung (z. B. Geschwindigkeit, Beschleunigung) von Körpern. In der Robotik sind Körper mithilfe von Gelenken zu kinematischen Ketten verbunden, die oft einfach mit „Kinematik“ bezeichnet werden.
Kinematikbetrieb, -betriebsart	Betriebsart, in welcher die Achsen der Kinematik durch die Kinematiksteuerung interpoliert werden. Im Gegensatz zur Steuerung von Einzelachsen werden bei der Profilgenerierung die anderen Achsen der Kinematik berücksichtigt.
Kinematikfehler	Ein im Kinematikbetrieb aufgetretener Fehler.
Kinematikfunktion	Funktion der Kinematiksteuerung, z. B. Linearinterpolation.
Kinematikinstanz	Instanz der Kinematiksteuerung. Mit einer Kinematikinstanz lässt sich eine Kinematik steuern.
Kinematikkoordinatensystem (KCS)	Koordinatensystem, das i. Allg. körperfest im Sockel der Kinematik positioniert ist. KCS ist in Bezug auf das Weltkoordinatensystem (WCS) definiert. Die Transformation, d. h. die Position des Roboters in der Welt, kann zur Laufzeit geändert werden.
Kinematiklimitierung	Begrenzung innerhalb der Kinematik. Der räumliche Arbeitsraum der Kinematik ergibt sich aus den Kinematiklimitierungen, den Begrenzungen der Achsen und den kartesischen Begrenzungen. Die konkrete Bedeutung einer Kinematiklimitierung hängt vom Kinematikmodell ab. Beispielsweise kann mithilfe Kinematiklimitierungen die Bewegung eines Gelenks einer Parallelkinematiken begrenzt werden, das nicht direkt durch einen Antrieb bewegt wird. Bei Überschreitung geht die Kinematiksteuerung i. Allg. in einen Fehlerzustand über.
Kinematikmodellachse	Achse mit Nullpunkt und Bewegungsrichtung gemäß Definition im bestimmten Kinematikmodell.
Kinematische Kette	Aneinanderreihung von Segmenten, die durch Gelenke miteinander verbunden sind.
Knickarmroboter	Knickarmroboter weisen 5 bis 6 Freiheitsgrade auf und umfassen mehrere Drehgelenke.
Konstellation (Constellation)	Stellung einer kinematischen Kette, mit der das TCS eine bestimmte Pose einnimmt. Im Allgemeinen können kinematische Ketten die gleiche Pose mit verschiedenen Stellungen, den sogenannten Konstellationen, erreichen. Beispielsweise kann das Ellbogengelenk einer SCARA-Kinematik auf die eine oder andere Seite eingeklappt sein, was zwei verschiedenen Konstellationen entspricht.
Koordinatenachse	Achse eines kartesischen Koordinatensystems.
Kreisinterpolation	Bestimmung der Bahn in Form eines Kreisbogens zwischen Anfangs- und Endpunkt.
Kreisumlauf	Bewegung entlang der kompletten Kreisbahn. (1 Kreisumlauf ≙ 360 Grad)
Krümmungssprung	Unstetigkeit der Bahnkrümmung als Funktion des Bahnfortschritts.

20211732 / DE – 07/2014

L	Bedeutung
Lastvorsteuerung	Reglerparameter zur Einstellung des Drehmoment-Sollwertes bei der Freigabe eines Antriebs, um das "Absacken" einer unter Last stehenden Achse zu verhindern.

M	Bedeutung
Mapping	Im Sinne einer Abbildung zu verstehen. Beispiel: Abbildung der aktuellen Orientierungskoordinaten in ABC-Darstellung auf Zielwerte.
Mixed Kinematik	Kinematikmodell, das nicht eindeutig die Charakteristiken anderer Kinematikmodelle aufweist. Dabei entspricht dieses insbesondere nicht den folgenden Modellen: Cartesian Gantry, Roller Gantry, SCARA, Delta, Quadropod, Hexapod oder Articulated.
Modellachsen	Achsen des Kinematikmodells im Ggs. zu den realen Achsen.

N	Bedeutung
Nullstellung des Kinematikmodells	Stellung des Kinematikmodells, in der sich alle Kinematikmodellachsen im Nullpunkt befinden.

O	Bedeutung
Orientierung	Rotatorische Ausrichtung eines Koordinatensystems. Wird angegeben durch eine Drehung im Verhältnis zu einem Bezugskoordinatensystem, z. B. in ABC-Darstellung.

P	Bedeutung
Packmuster	Anordnung von Objekten, z. B. Paketen, auf einer Palette.
Palettieren	Zusammenfassen von Packstücken (z. B. Kisten) auf einem Ladungsträger (z. B. Palette) mit Hilfe eines Packmusters.
Parallelkinematik	Kinematikmodell, bei dem die kinematische Kette zwischen Basiskoordinatensystem (BCS) und Flanschkoordinatensystem (FCS) aus einer Parallelschaltung von mindestens zwei kinematischen Ketten besteht.
Pick-and-Place	Aufnehmen und Platzieren von Objekten.
Pose	Position (X, Y, Z) und Orientierung (A, B, C) des Werkzeugkoordinatensystem (TCS). Oft einfach mit "Position" bezeichnet.

Q	Bedeutung
Quadropod-Kinematik	Parallelkinematik, die durch 4 parallele kinematische Ketten charakterisiert ist.

R	Bedeutung
Rast-in-Rast	Bewegungsablauf, bei der die Bewegung aus dem Stillstand startet und im Stillstand endet (z. B. bei Pick-and-Place-Anwendungen) - im Gegensatz zu einem Bewegungsablauf, bei dem sich ein bereits bewegtes Objekt auf die Bewegung eines anderen Objekt aufsynchronisiert.
Raumkoordinate	Kartesische Koordinate
Roboterflansch	Ort am Roboter für die Montage des Werkzeugs

20211732 / DE – 07/2014

R	Bedeutung
Robotersockel	Basissegment eines Roboters, das an der Umgebung (z. B. Boden, Wand, Decke) befestigt sein kann oder z. B. auf einer Linearachse verfahren wird.
Roller Gantry	Kinematikmodell, bei dem i. Allg. zwei translatorische Freiheitsgrade von zwei i. Allg. stationären Antrieben über einen umlaufenden Zahnriemen gesteuert werden. Dieser Baugruppe können weitere Bewegungsfreiheitsgrade vor- und/oder nachgelagert sein.
Rotation	Drehung; Veränderung der Orientierungskoordinaten A, B, C z. B. des Werkzeugkoordinatensystems (TCS) der Kinematik

S	Bedeutung
SCARA-Kinematik	engl.-sprachige Abkürzung für **S**elective **C**ompliance **A**ssembly **R**obot **A**rm. Kinematische Kette, welche ähnlich dem menschlichen Arm durch ein Schulter- und ein Ellbogengelenk charakterisiert ist, wobei es sich in der Robotik um zwei Drehgelenke mit zueinander parallelen, rotatorischen Achsen handelt.
Stützpunkttabelle	Liste von Punkten, die eine Bahn beschreiben.

T	Bedeutung
TARGET-Betriebsart	Kinematikbetriebsart, die durch das unabhängige oder synchronisierte Positionieren von Achsen oder kartesischen Koordinaten auf Zielwerte charakterisiert ist.
Teleskopachse	Glied einer kinematischen Kette, deren Länge variabel ist.
Teleskoparm	Roboterarm mit integrierter Teleskopachse.
Tracking	Verfolgung eines bewegten (virtuellen oder realen) Objekts mit dem Werkzeug der Kinematik. Beispiele für typische Tracking-Anwendungen sind das Pick-and-Place von Objekten auf bewegte Transportbänder oder weg davon, sowie die Bearbeitung von Objekten, die sich in bis zu 6 Freiheitsgraden in Bewegung befinden. Hinweis: Kinematische Tracking-Anwendungen sind nicht zu verwechseln mit der MultiMotion-Betriebsart "Tracking", die durch die Kinematiksteuerung nicht verwendet wird.
Transition	Übergang zwischen zwei Zuständen.
Translation	Verschiebung; Veränderung der Positionskoordinaten X, Y, Z z. B. des Werkzeugmittelpunkts (TCP) der Kinematik.
Translationsgeschwindigkeit	Geschwindigkeit, mit der sich die Position z. B. des Werkzeugmittelpunkts (TCP) ändert.
Tripod-Kinematik	Parallelkinematik, die durch ein Dreibein charakterisiert ist.
Tupel	Ein Tupel ist eine Liste endlich vieler Objekte mit definierter Reihenfolge.

U	Bedeutung
Überschleifbereich	Räumlicher Bereich, in dem das Überschleifen von einem Bahnsegment auf das nächste Bahnsegment erfolgt.
Überschleifbewegung	Bewegung im Überschleifbereich bzw. auf dem Überschleifbogen.
Überschleifbogen	Bahnsegment im Überschleifbereich.

20211732 / DE – 07/2014

U	Bedeutung
Überschleifen	Stetiger Übergang des Bahnverlaufs sowie der Bahngeschwindigkeit auf das nächste Bahnsegment. Bei Continuous-Path-Interpolation kann die Bahngeschwindigkeit im Überschleifbereich optional auch konstant gehalten werden.
USER-Kinematik	Vom Standard abweichendes Kinematikmodell (z. B. Quadropod, Hexapod, Articulated, None etc.) bis hin zu Spezialkonstruktionen.

W	Bedeutung
Weltkoordinatensystem (WCS)	Stationäres Koordinatensystem mit der Erde als Bezug, das unabhängig von der Bewegung der Kinematik ist. Das Weltkoordinatensystem (WCS) ist das Bezugskoordinatensystem für das Kinematikkoordinatensystem (KCS) und das Werkzeugkoordinatensystem (PCS1/2). Das Weltkoordinatensystem (WCS) kann beispielsweise an einem Transportband platziert werden, von dem mehrere Roboter Objekte entnehmen. Die Orientierung des Weltkoordinatensystems (WCS) kann in Bezug auf das Gravitationskoordiantensystem (GCS) konfiguriert werden für korrekte Kraft-/Momentenberechnungen sowie für das Ausrichten in der 3D-Simulation.
Werkstückkoordinatensystem (PCS1/2)	Koordinatensystem an einem Werkstück, z. B. Palette oder auf Fließband transportiertes Objekt. PCS1/2 ist in Bezug auf das Weltkoordinatensystem (WCS) definiert. Die Transformation, d. h. die Position des Werkstücks, kann zur Laufzeit geändert werden.
Werkzeugkoordinatensystem (TCS)	Koordinatensystem im Werkzeug der Kinematik. Das Werkzeugkoordinatensystem (TCS) ist in Bezug auf das Flanschkoordinatensystem (FCS) definiert. Die Transformation von FCS zu TCS kann zur Laufzeit geändert werden.
Werkzeugmittelpunkt (TCP)	Ursprung des Werkzeugkoordinatensystems (TCS) der Kinematik.
Werkzeugtransformation	Koordinatentransformation von dem Flanschkoordinatensystem (FCS) zu dem Werkzeugkoordinatensystem (TCS).
Werkzeugwechsel	Wechsel des Werkzeugs, das am Roboterflansch befestigt ist.

Z	Bedeutung
Zielpunkt	Punkt, zu dem der Werkzeugmittelpunkt (TCP) der Kinematik verfahren werden soll.
Zielvorgabe	Vorgabe der Zielposition und der Zielorientierung.
Zwischenpunkt	Endpunkt eines Bahnsegments, das noch nicht das letzte Segment der Bahn ist.

16.2 Abkürzungsverzeichnis

Abkürzung	Begriff	
	EN	DE
ACS	Axis Coordinate System	Achskoordinatensystem
BCS	Base Coordinate System	Basiskoordinatensystem
CP	Continuous Path	-
DOF	Degree of Freedom	Freiheitsgrad

20211732 / DE – 07/2014

Abkürzung	Begriff	
	EN	DE
FCP	Flange Center Point	Flanschmittelpunkt
FCS	Flange Coordinate System	Flanschkoordinatensystem
GCS	Gravity Coordinate System	Gravitationskoordinatensystem
KCS	Kinematics Coordinate System	Kinematikkoordinatensystem
PCS1/2	Piece Coordinate System	Werkstückkoordinatensystem
SCARA	Selective Compliance Assembly Robot Arm	-
TCP	Tool Center Point	Werkzeugmittelpunkt
TCS	Tool Coordinate System	Werkzeugkoordinatensystem
WCS	World Coordinate System	Weltkoordinatensystem

20211732 / DE – 07/2014

Stichwortverzeichnis

A

Abschnittsbezogene Sicherheitshinweise 7
Applikationsmodule
  Diagnose 41

B

Bussystem 11

E

Eingebettete Sicherheitshinweise 8

G

Gefahr
  Wechsel in den Steuermodus 44
Gefahrensymbole
  Bedeutung 8

H

Haftung 8
Haftungsausschluss 9
Hinweise
  Bedeutung Gefahrensymbole 8
  Kennzeichnung in der Dokumentation 7

I

Inbetriebnahme
  Ablauf 25
  Voraussetzungen 26

K

Konfiguration
  Neu erstellen 30

M

Mängelhaftung 8
Marken 9
mitgeltende Unterlagen 9
Monitor
  -betrieb 43
MotionStudio
  Handhabung 26
MOVITOOLS®, siehe MotionStudio 26

P

Produktnamen 9

S

Sachmängelhaftung 9
Selbsttätiger Anlauf der Maschine 44
Sicherheitshinweise 10
  Allgemeine 10
  Aufbau der abschnittsbezogenen 7
  Aufbau der eingebetteten 8
  Kennzeichnung in der Dokumentation 7
Signalworte in Sicherheitshinweisen 7
Start
  Diagnose 41
  Konfiguration 30
Steuerbetrieb 43

U

Unterlagen, mitgeltende 9
Urheberrechtsvermerk 9

V

Verwendung, bestimmungsgemäße 11
Voraussetzungen
  Inbetriebnahme 26

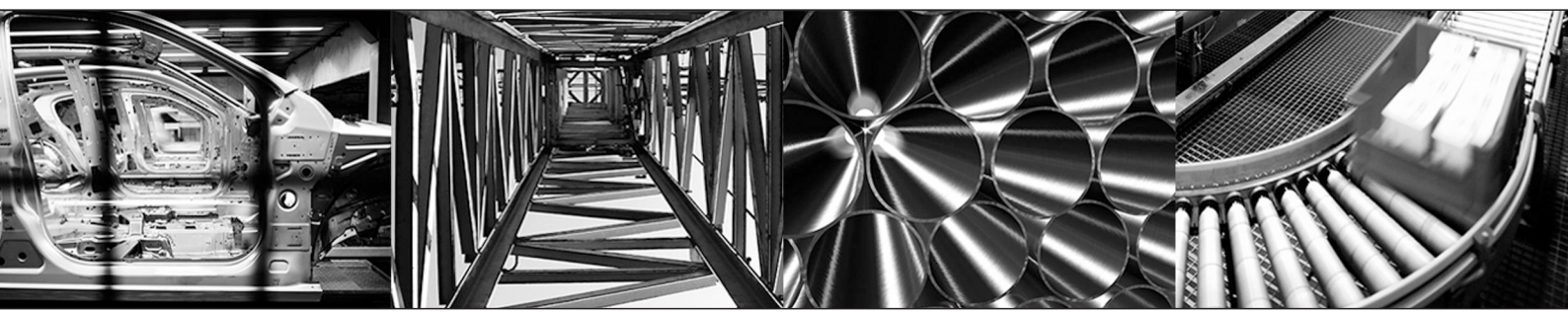
W

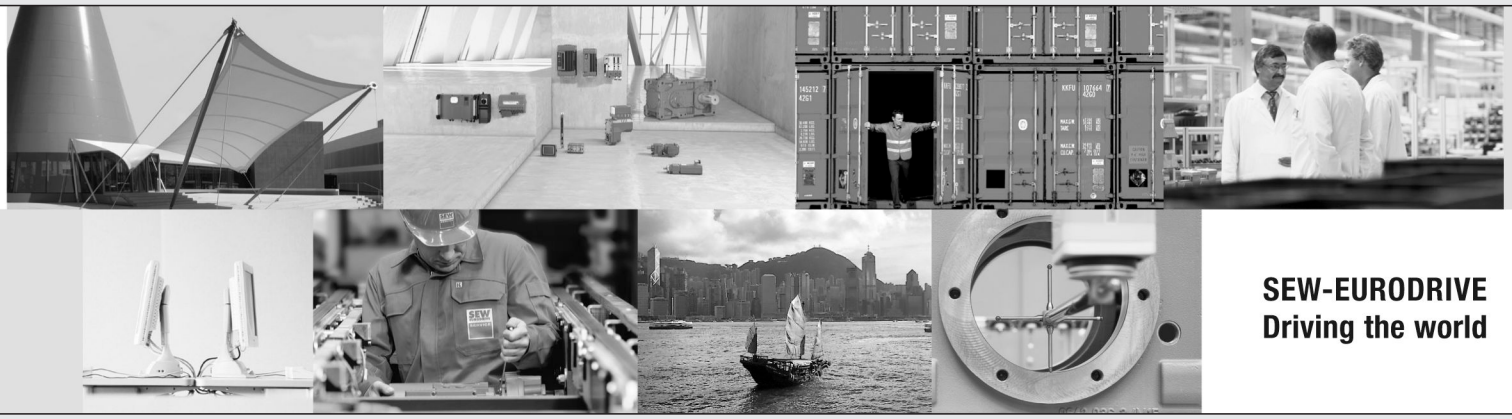
Warnhinweise
  Bedeutung Gefahrensymbole 8

Z

Zielgruppe 10

20211732 / DE – 07/14





SEW-EURODRIVE
Driving the world

SEW-EURODRIVE GmbH & Co KG
P.O. Box 3023
76642 BRUCHSAL
GERMANY
Phone +49 7251 75-0
Fax +49 7251-1970
sew@sew-eurodrive.com
➞ **www.sew-eurodrive.com**